

FIG. 1

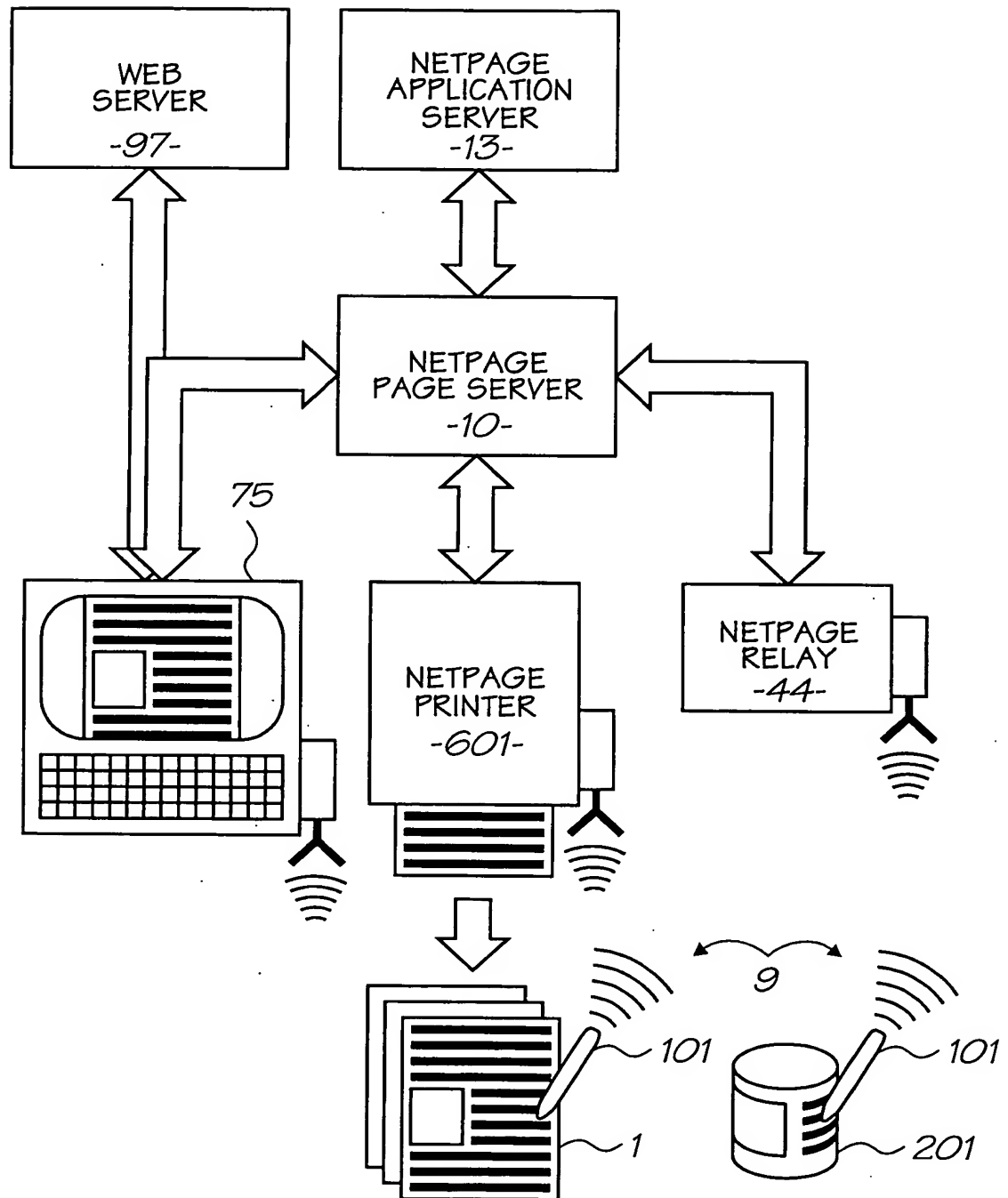


FIG. 2

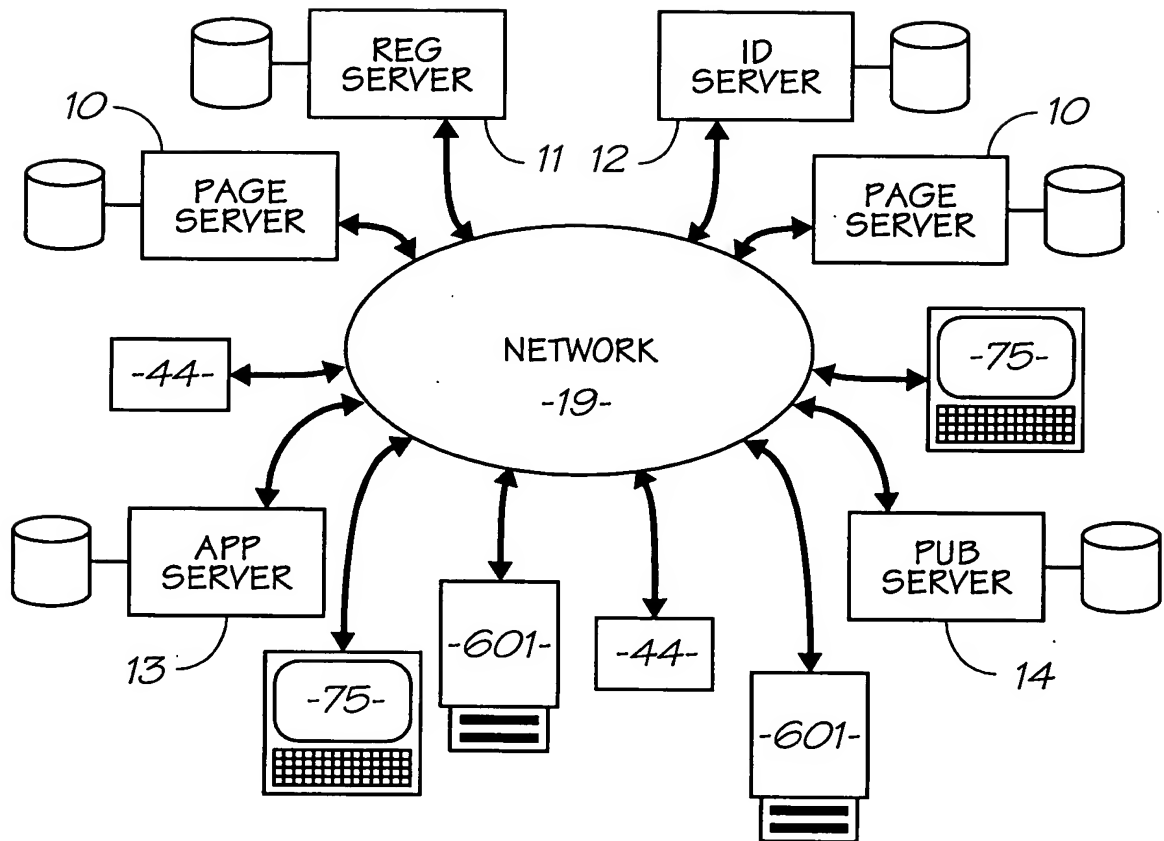


FIG. 3

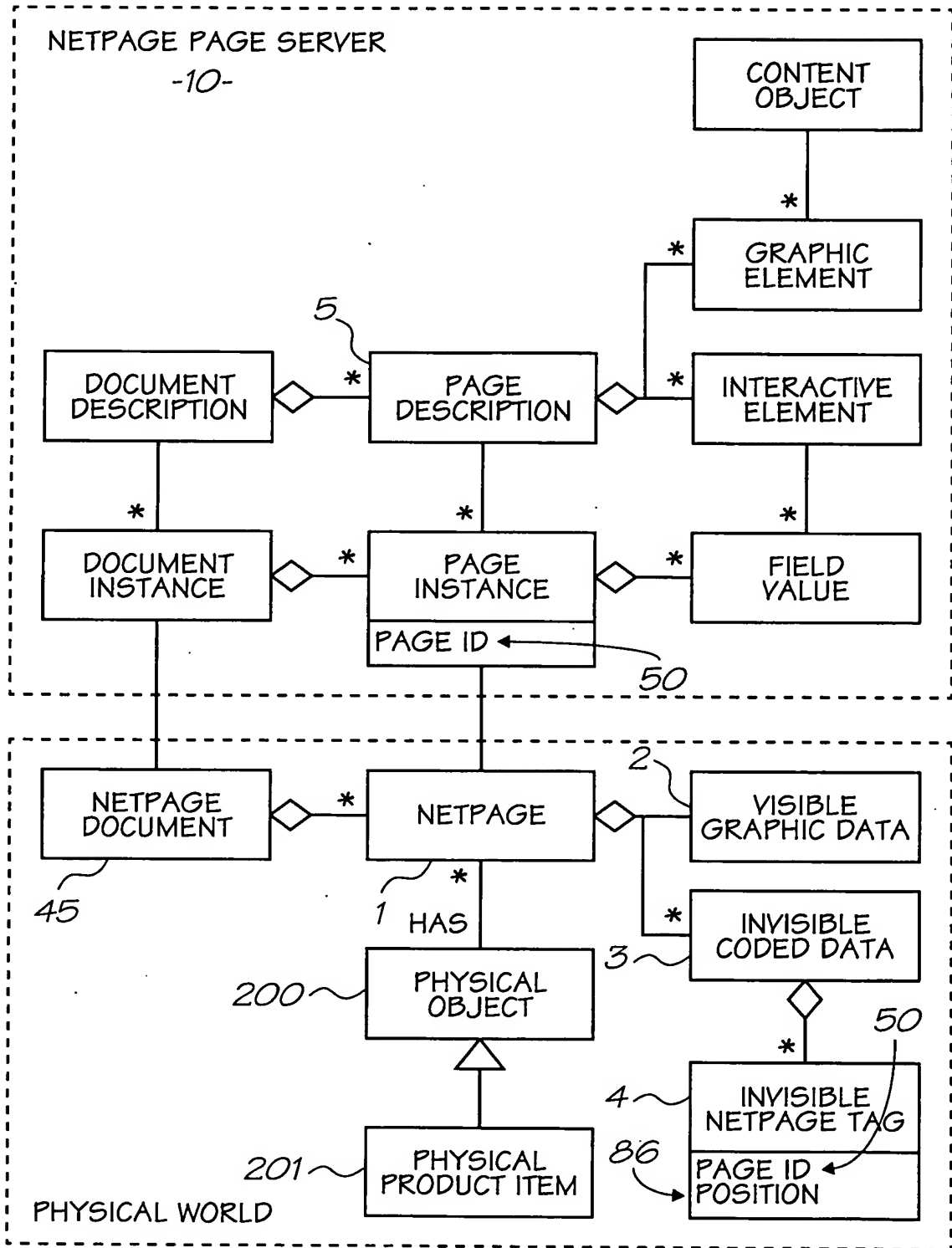


FIG. 4



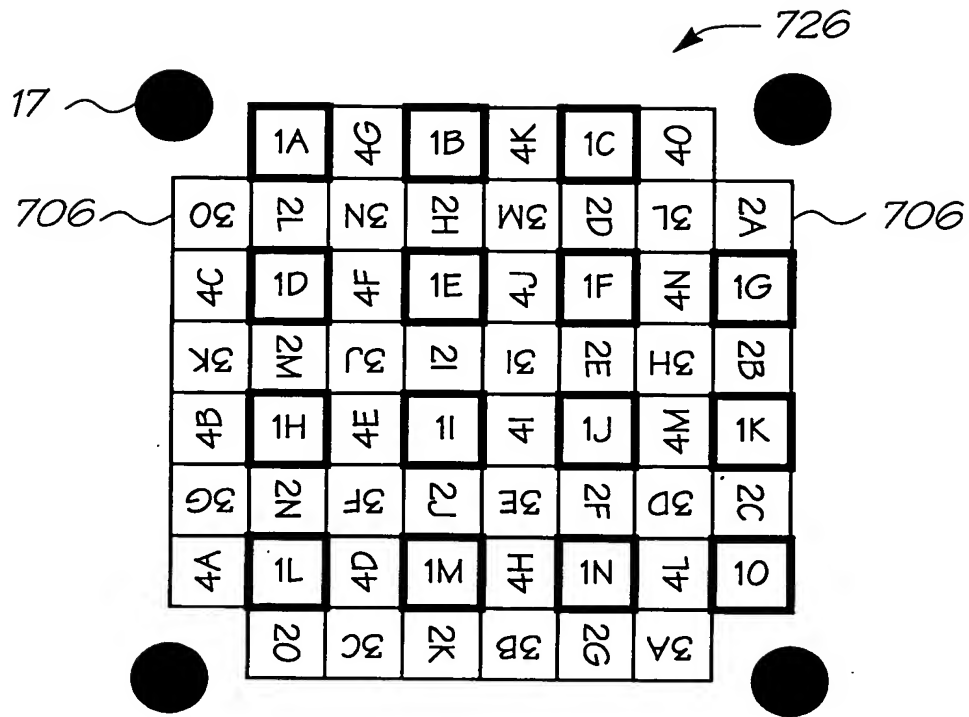


FIG. 5a

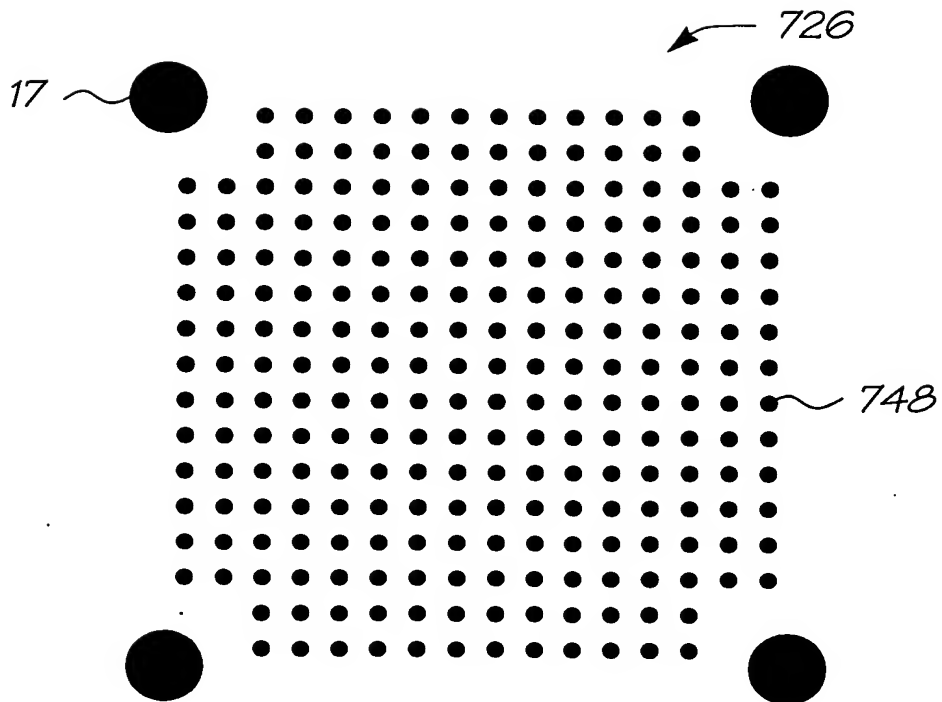
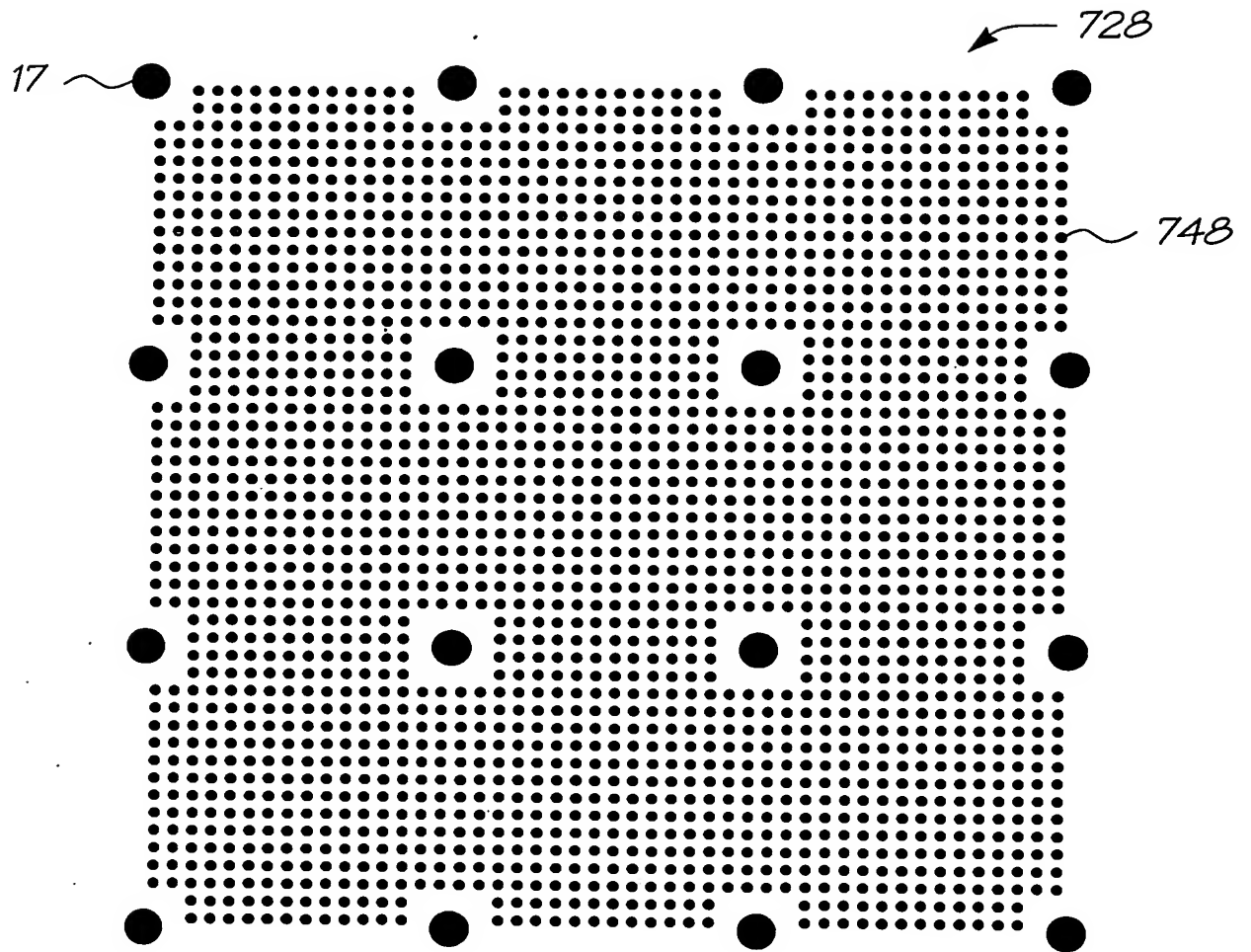


FIG. 5b

*FIG. 5c*

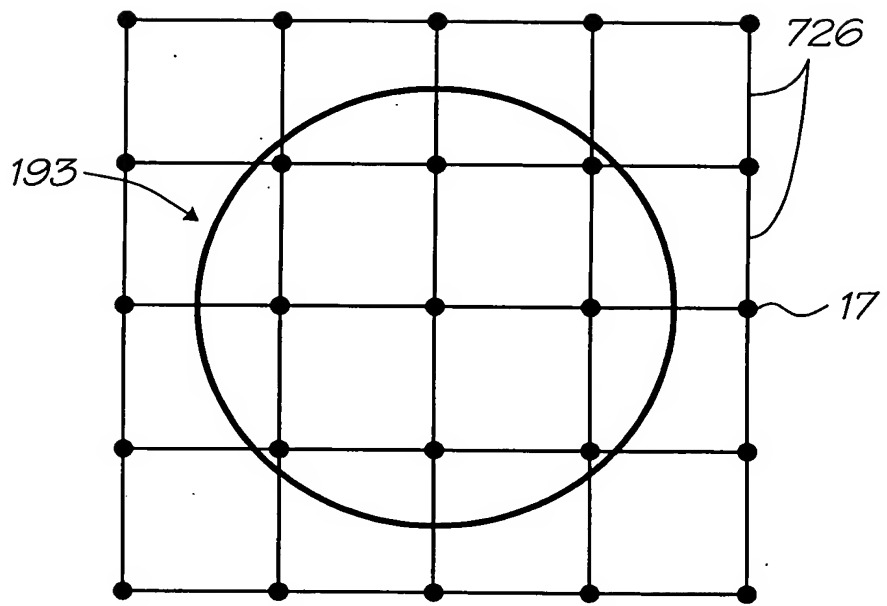


FIG. 6

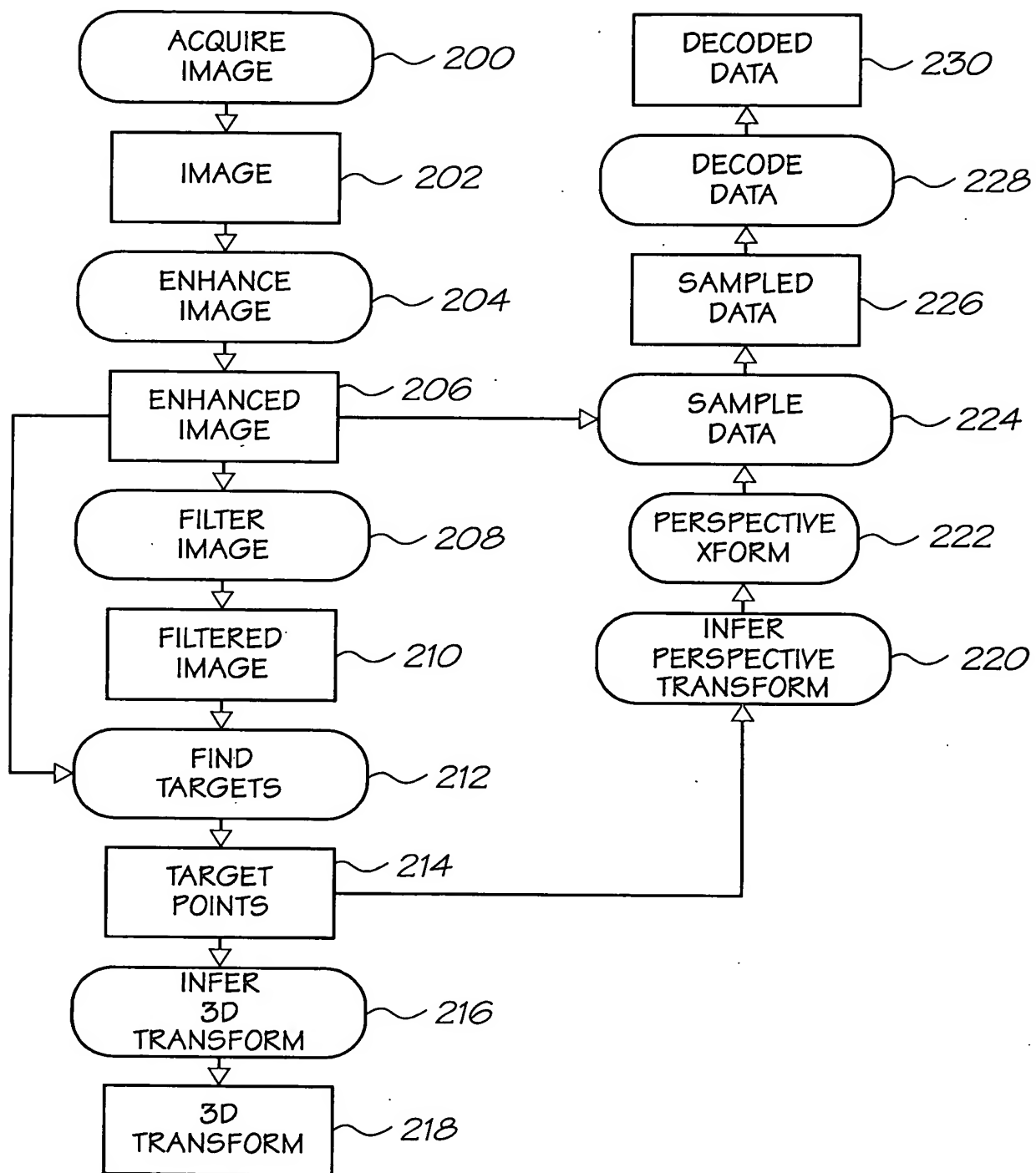


FIG. 7

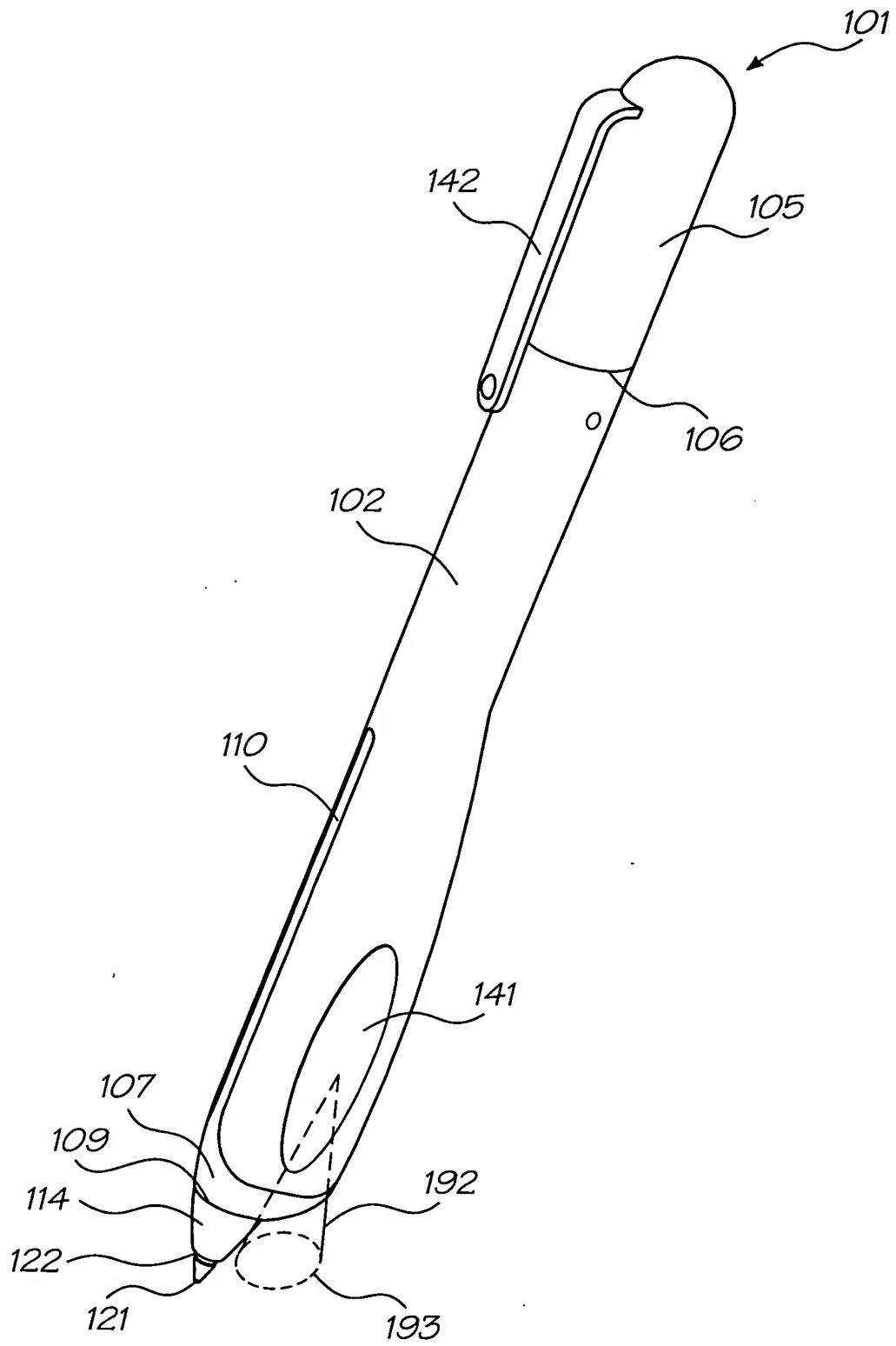


FIG. 8

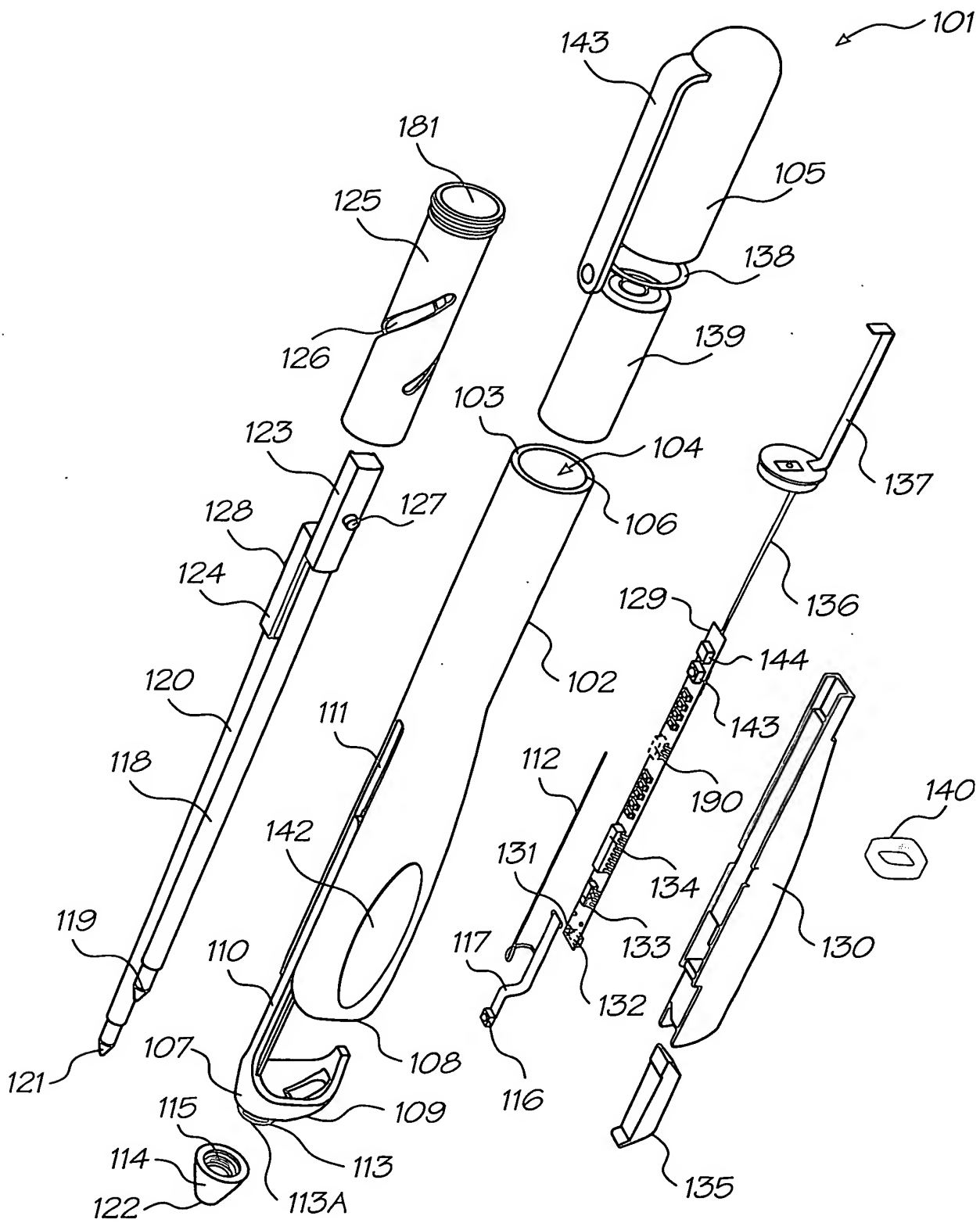


FIG. 9

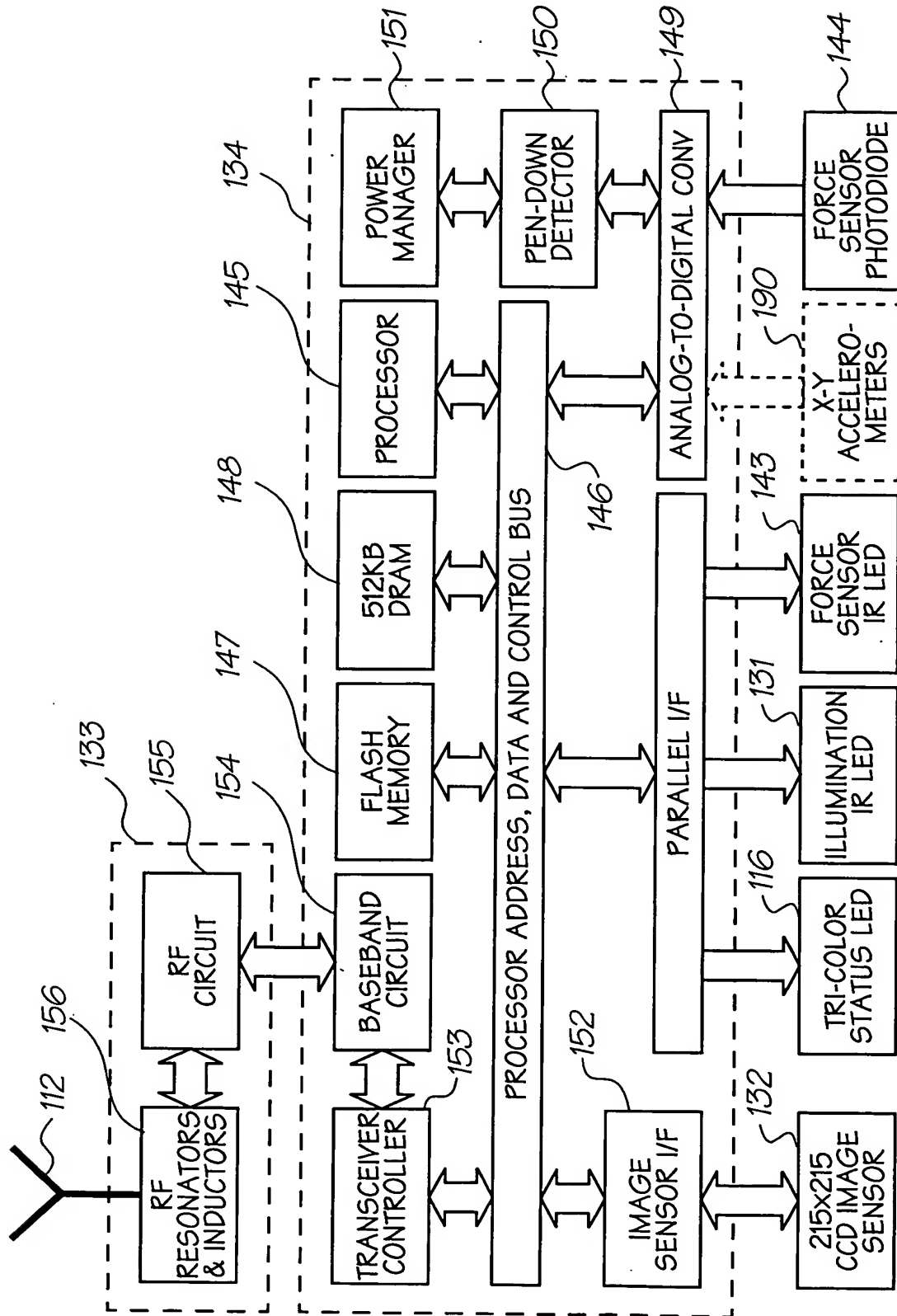


FIG. 10

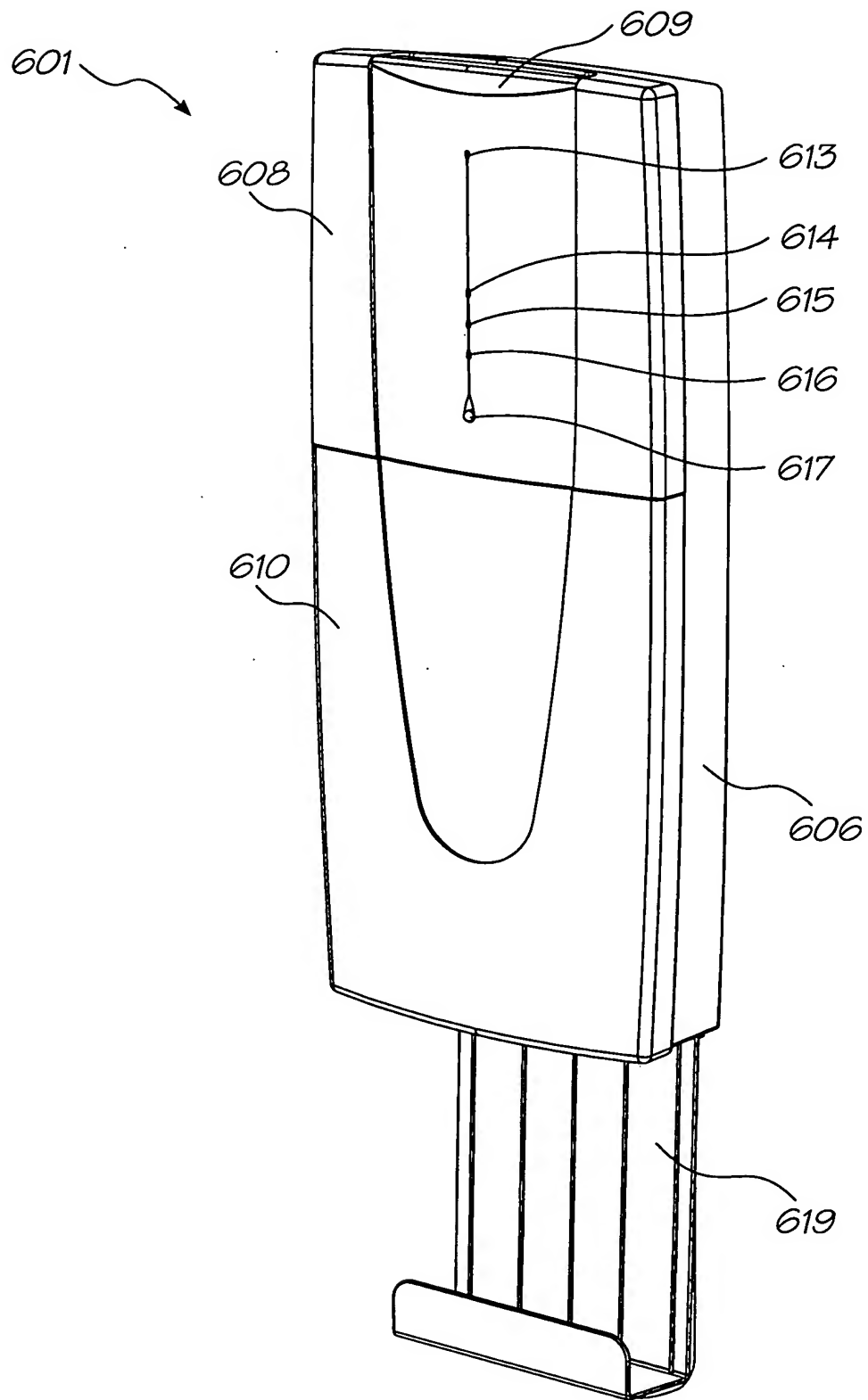


FIG. 11



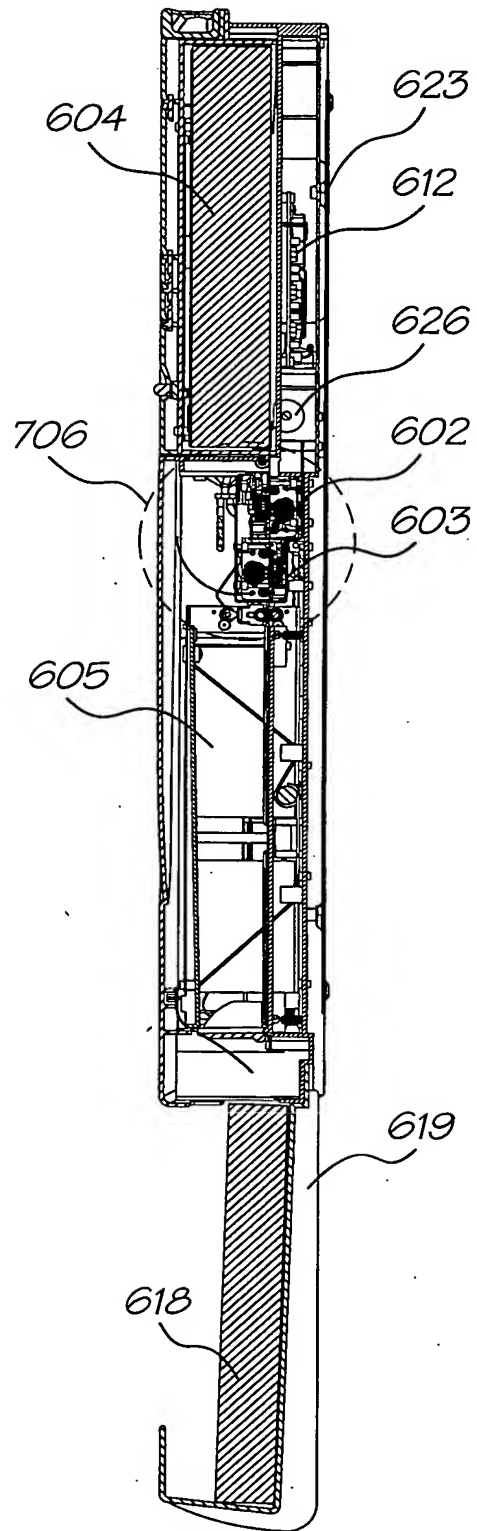


FIG. 12

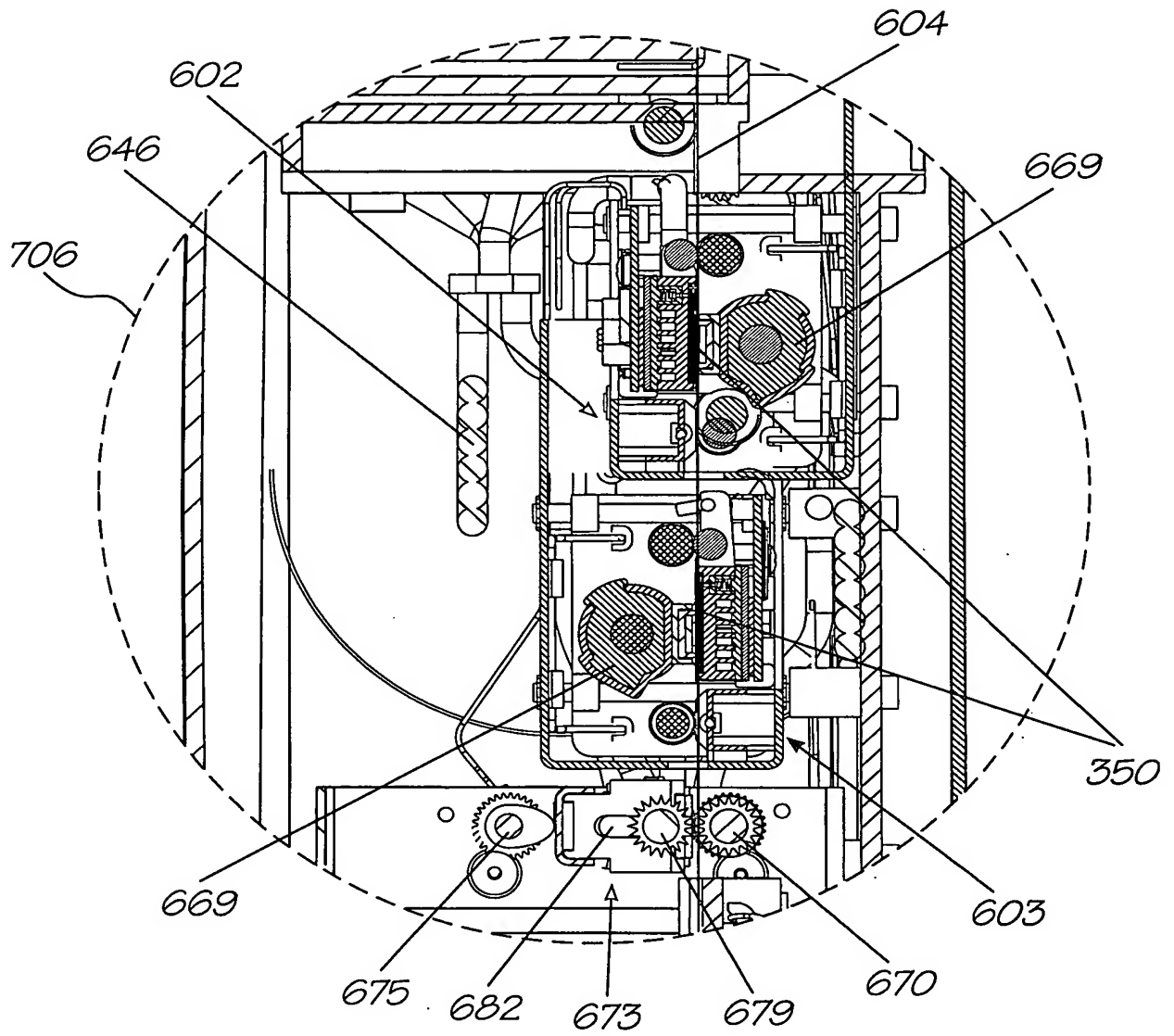


FIG. 12a

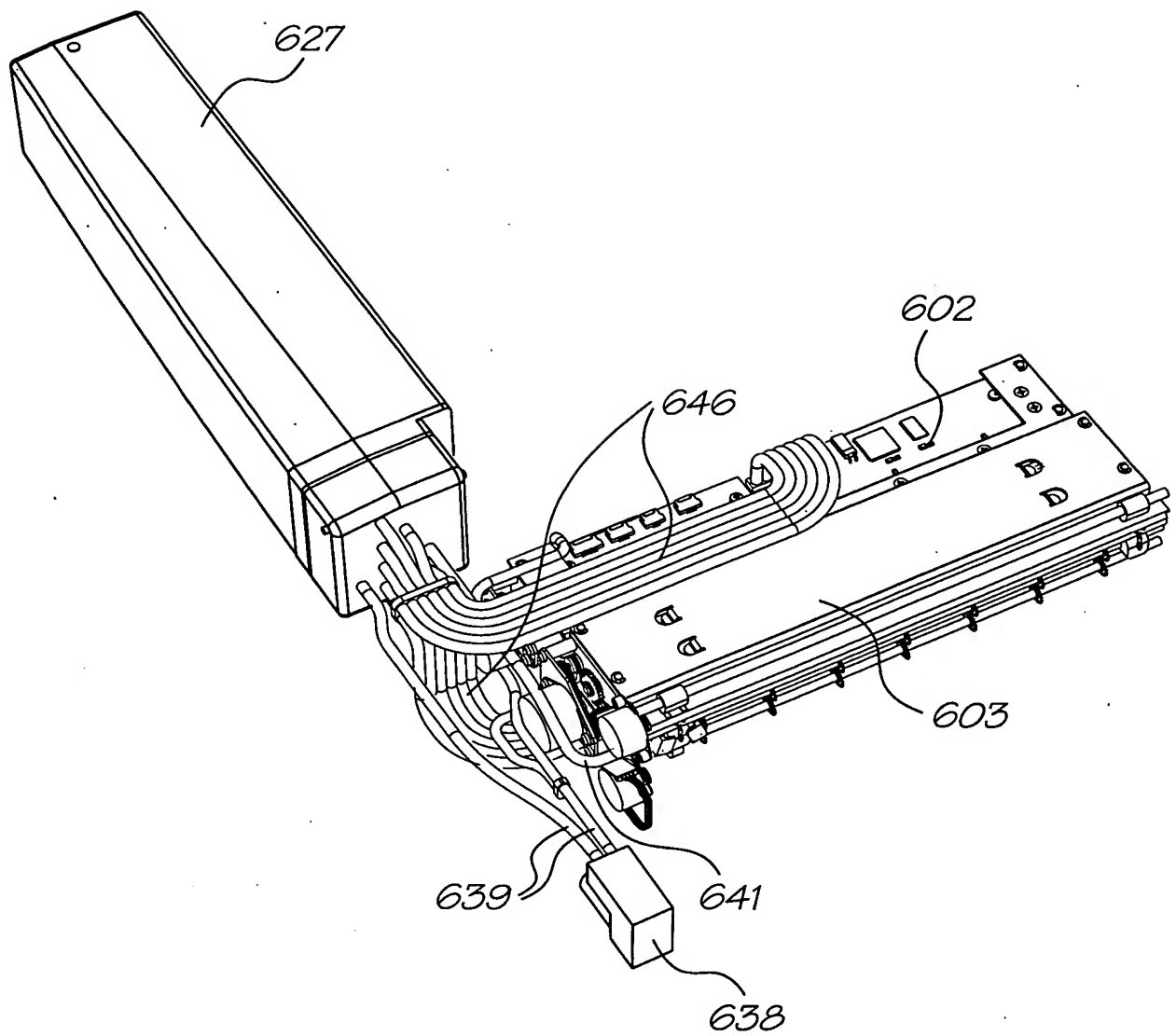


FIG. 13

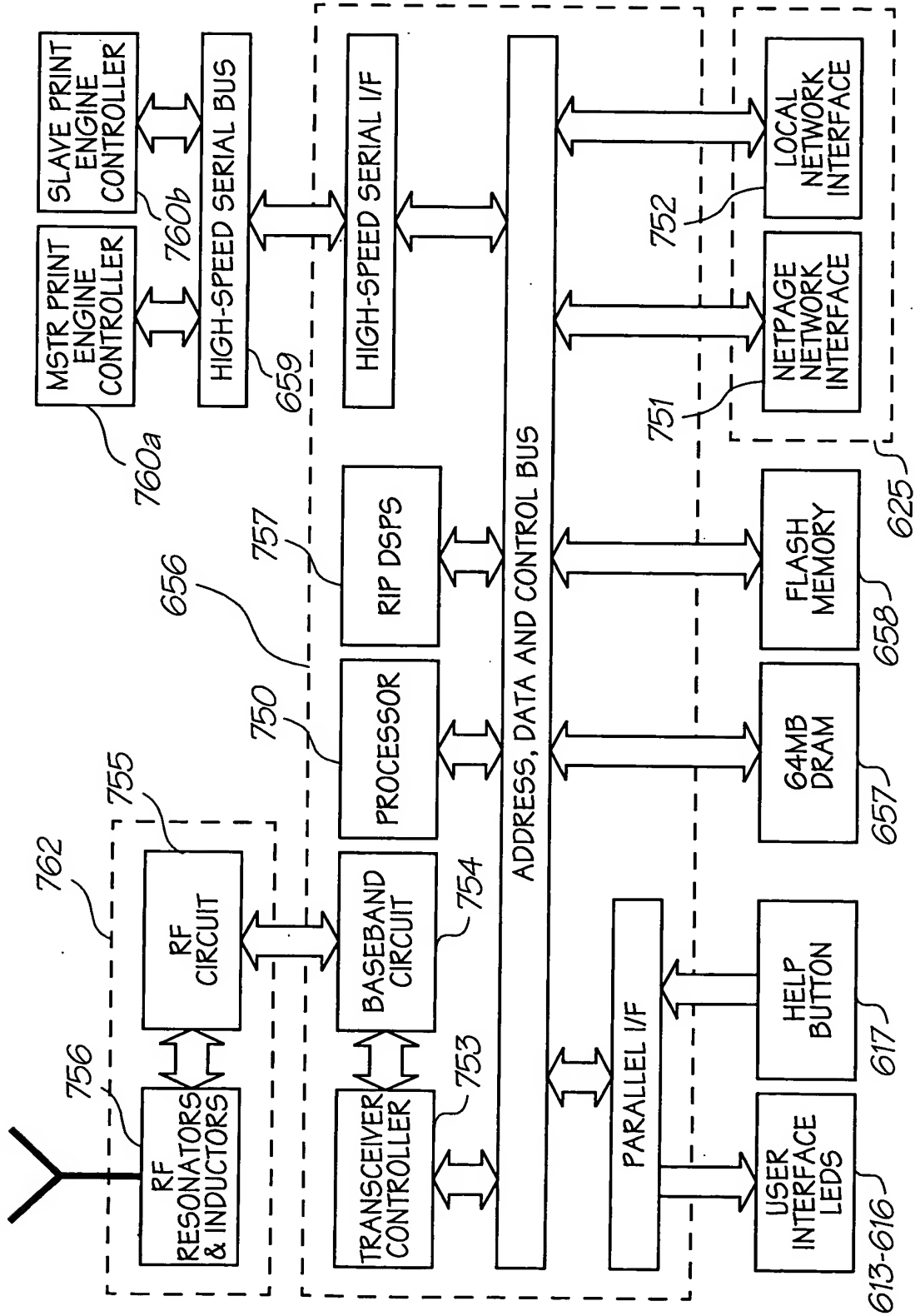


FIG. 14

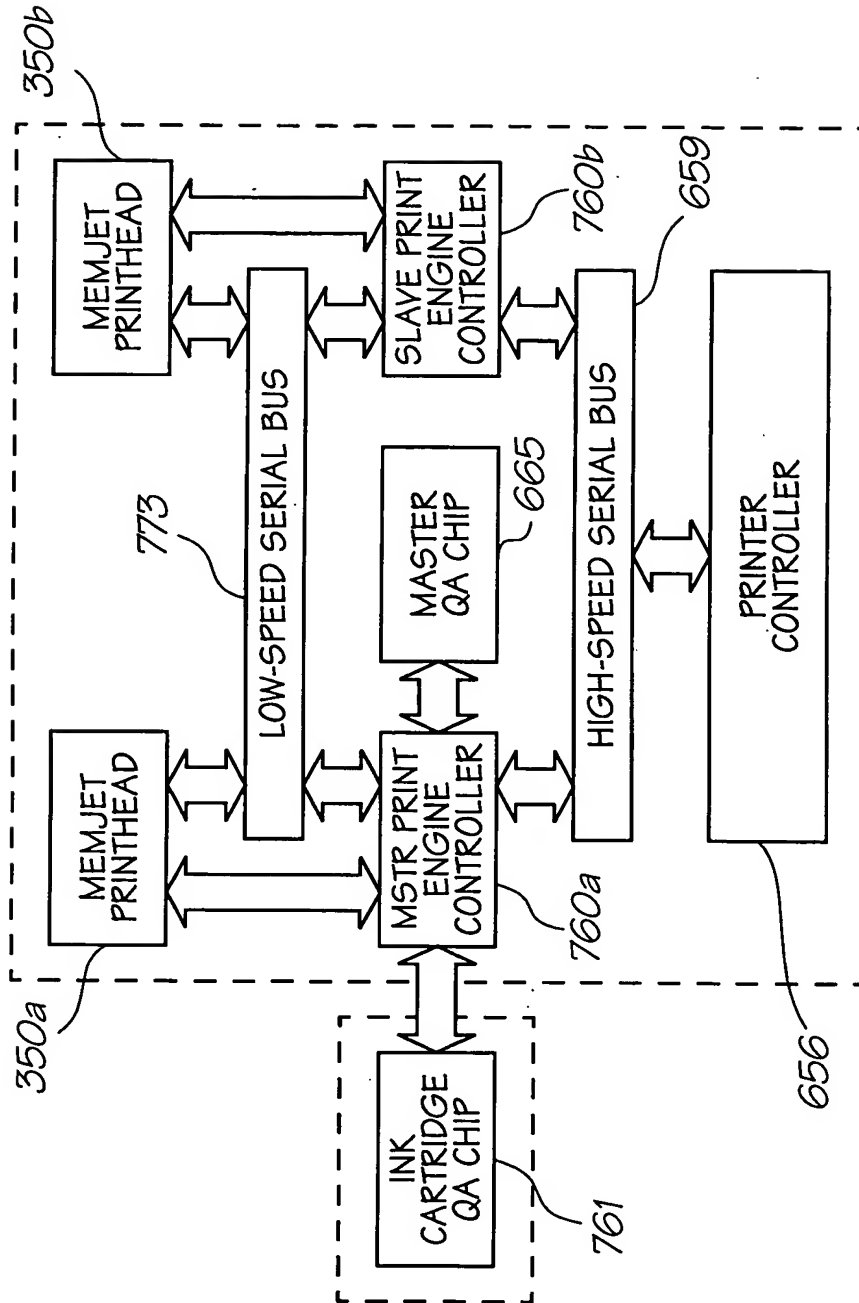


FIG. 15

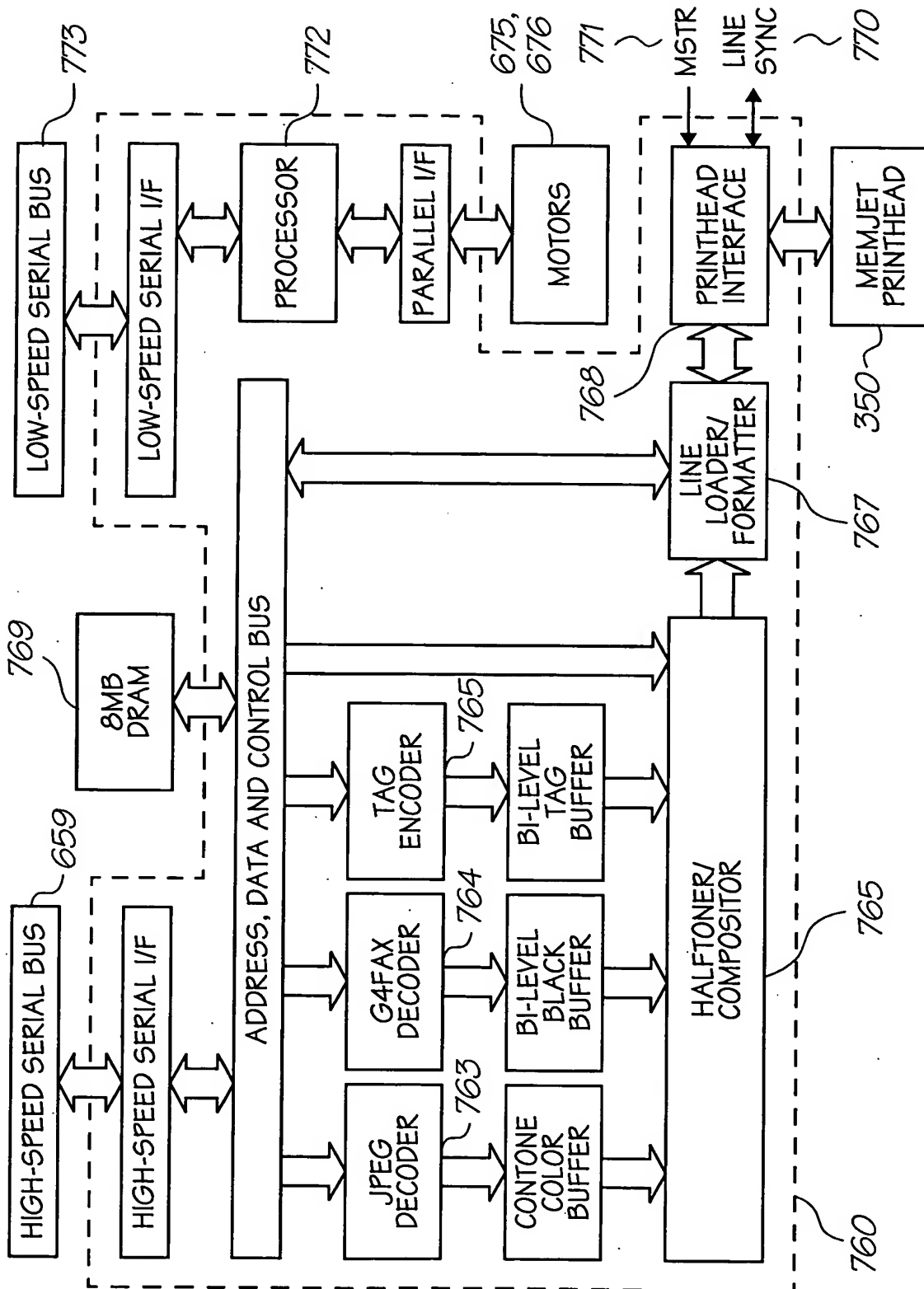


FIG. 16

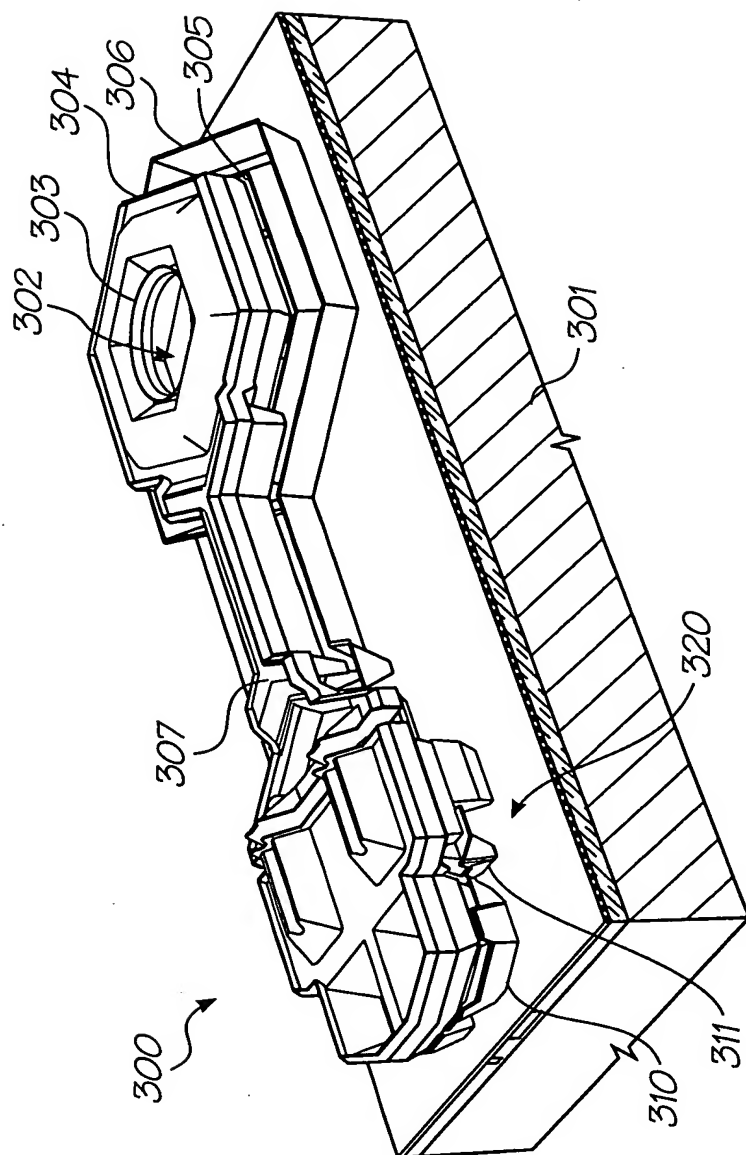


FIG. 17

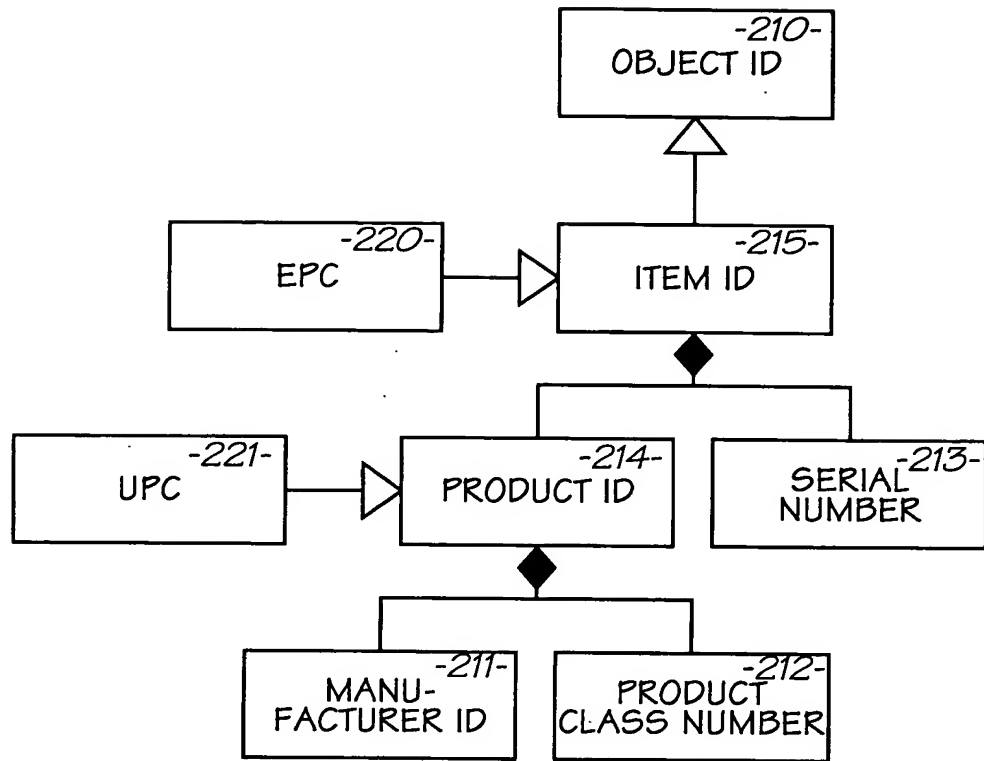


FIG. 18

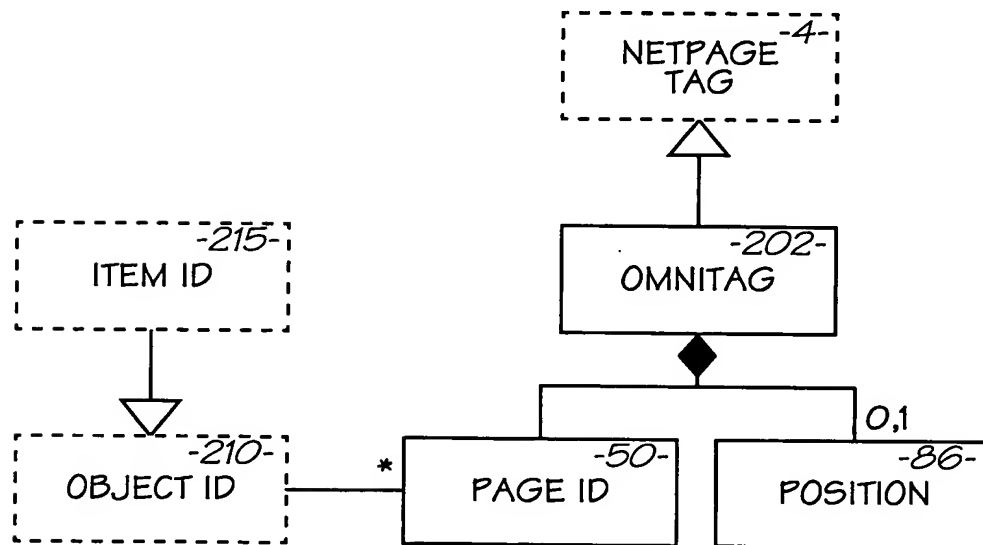


FIG. 19



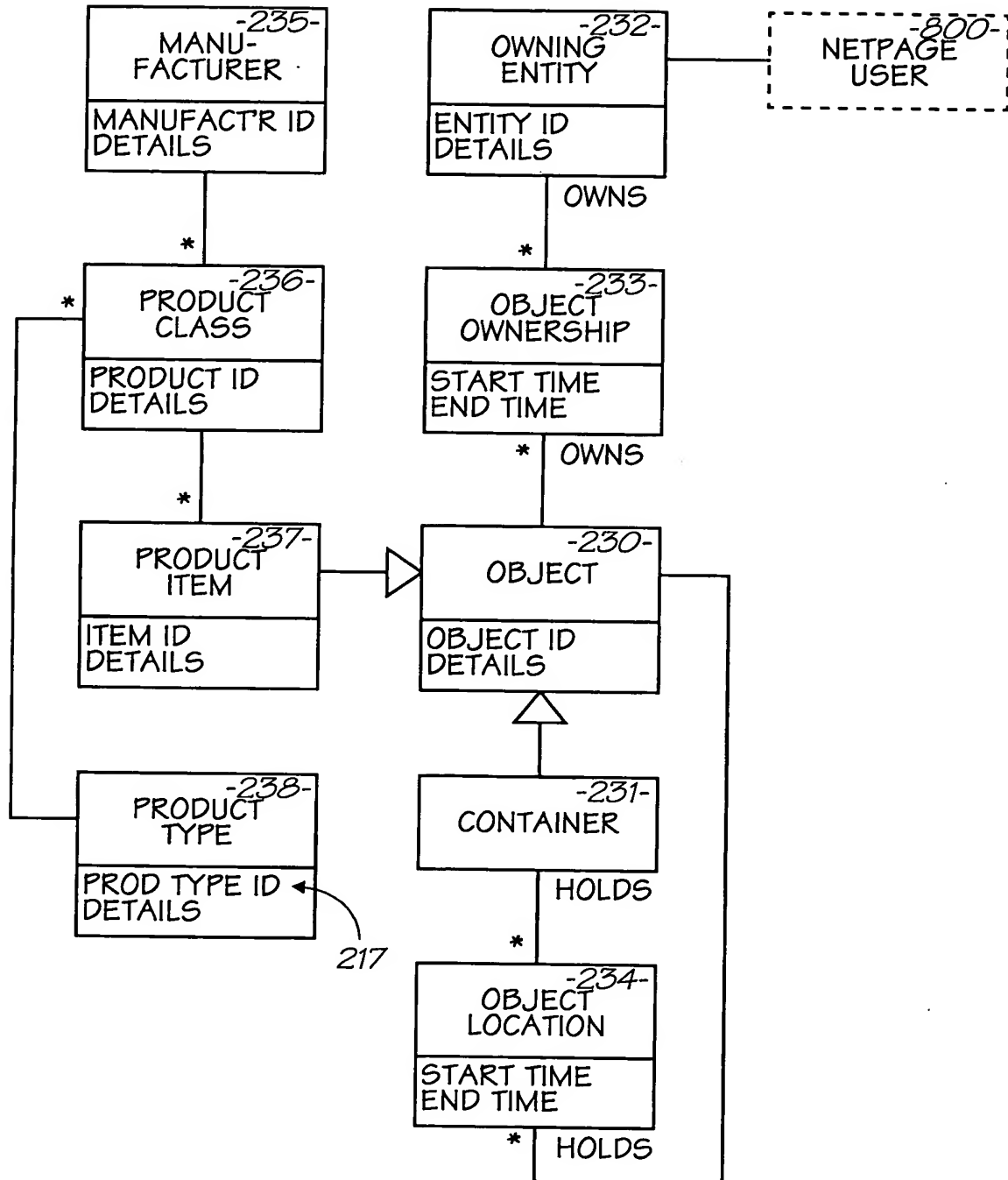


FIG. 20

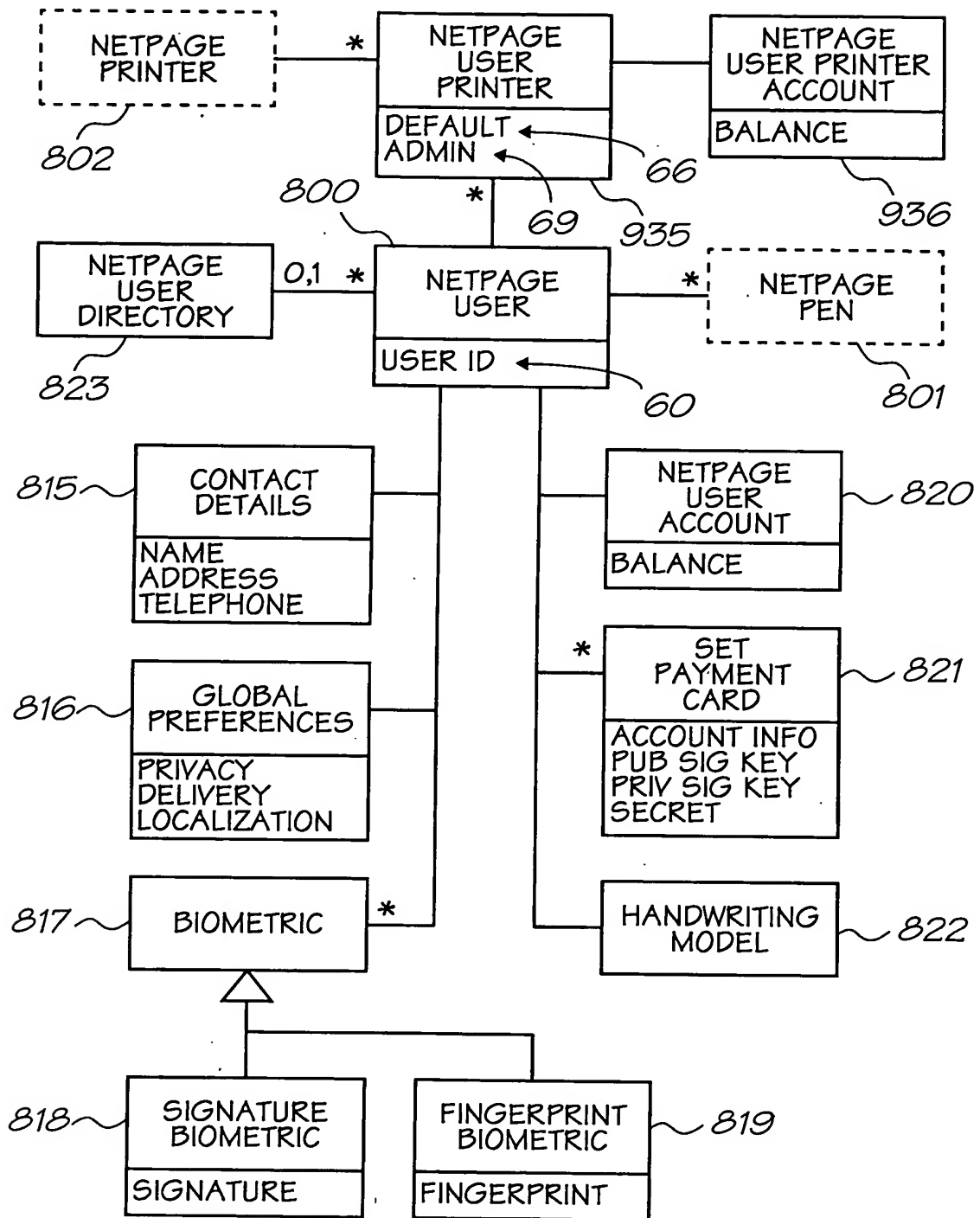


FIG. 21

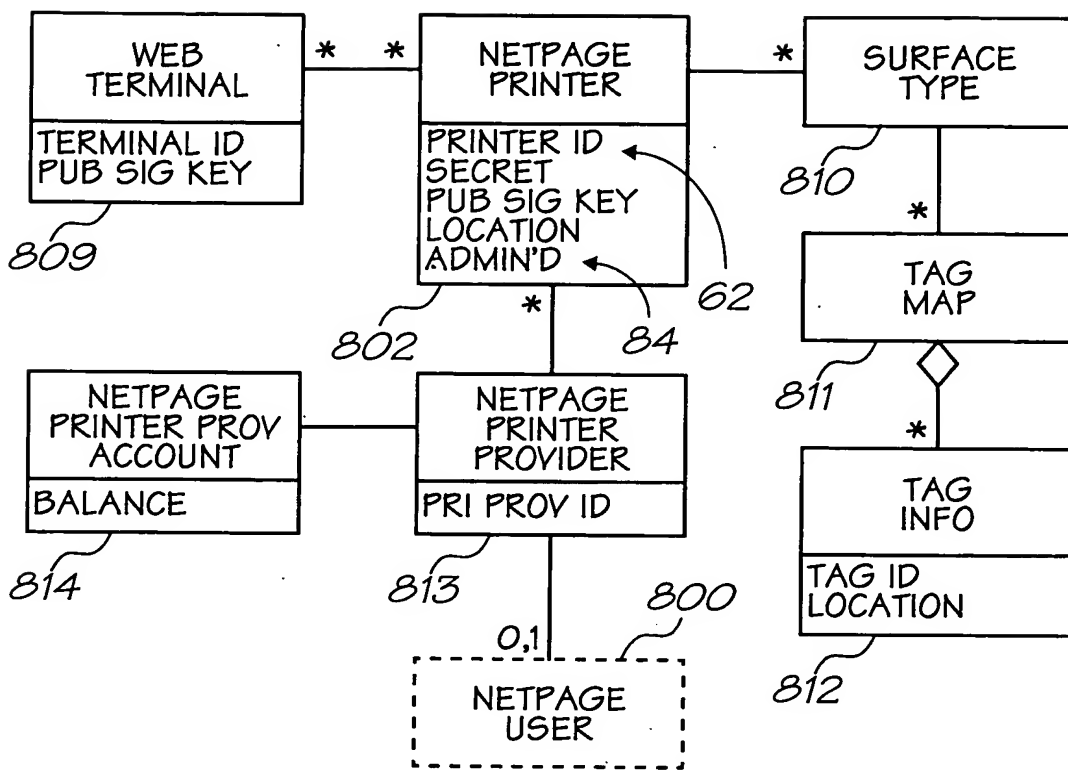


FIG. 22

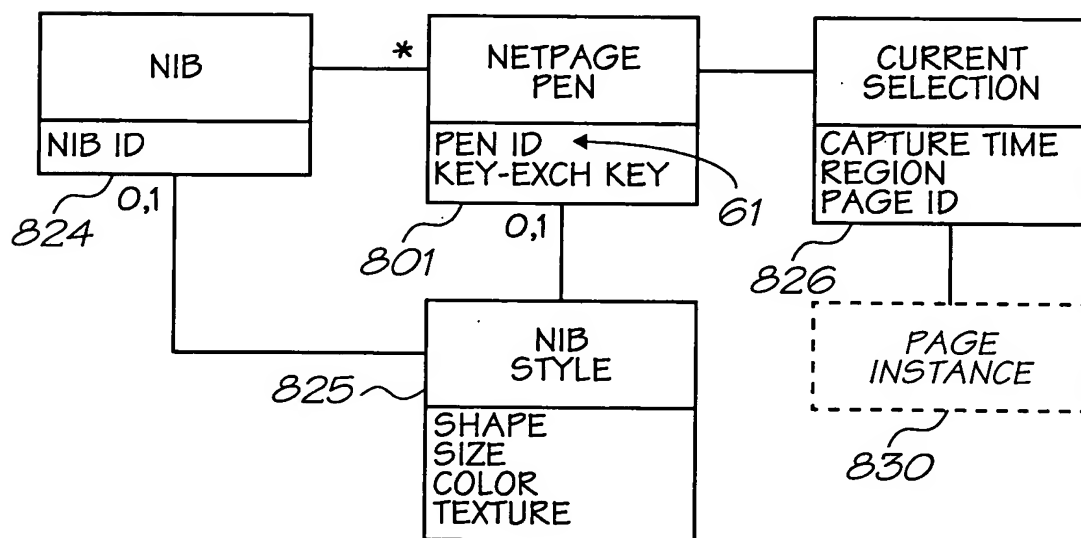


FIG. 23

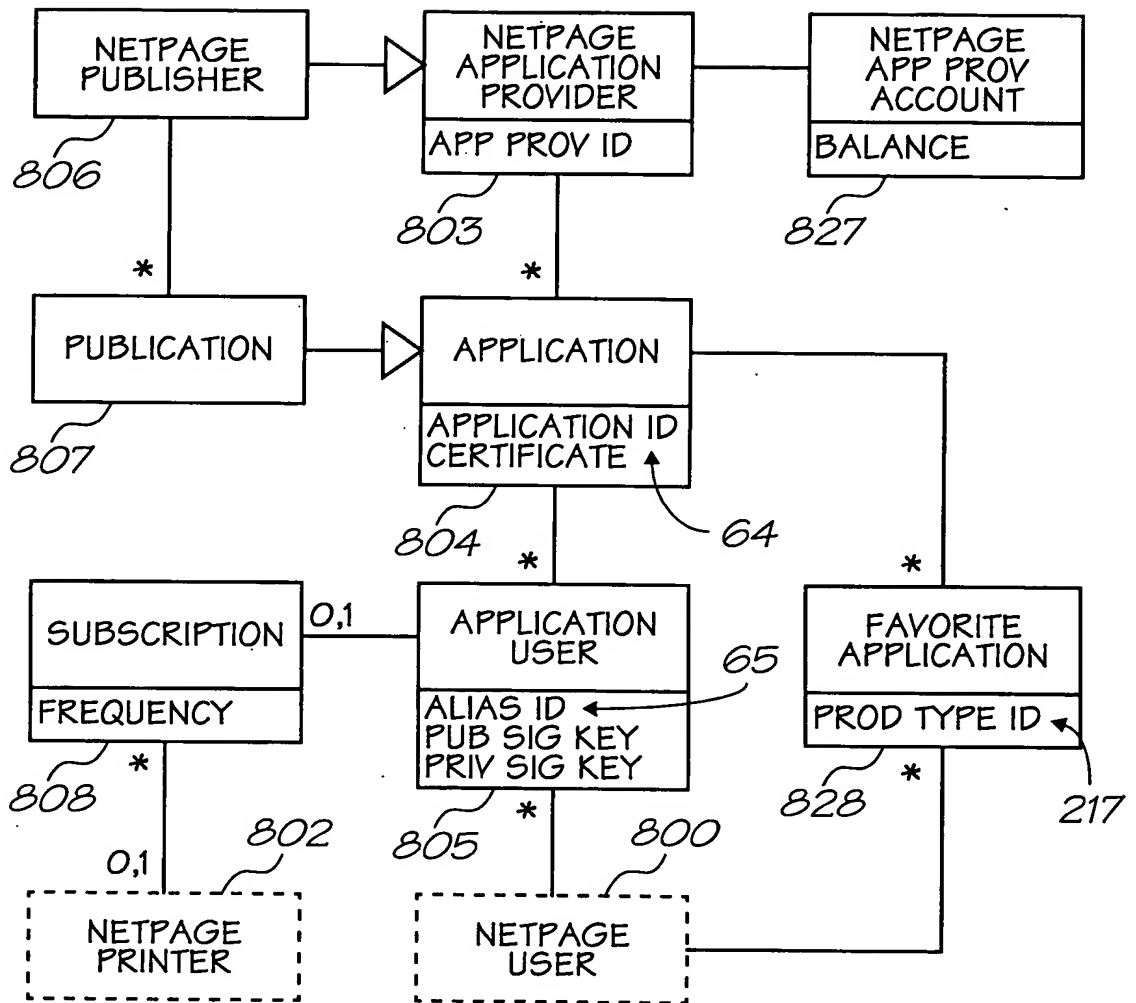


FIG. 24

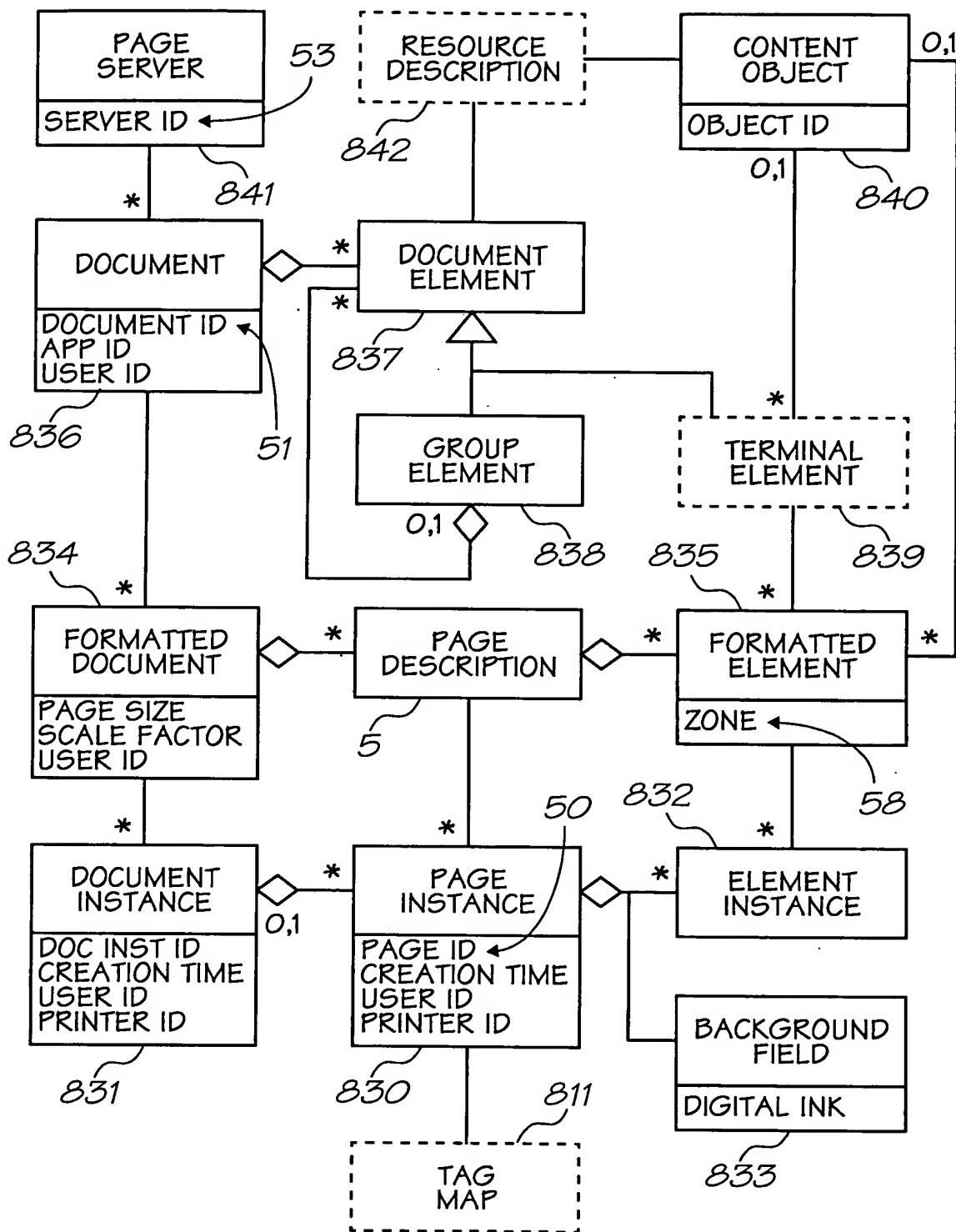


FIG. 25

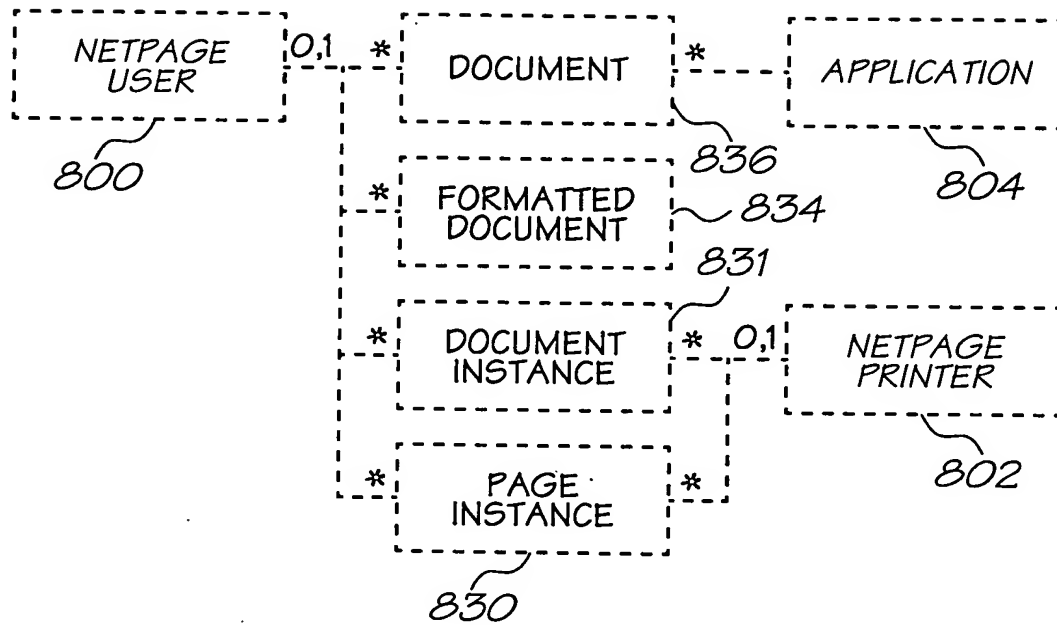


FIG. 26

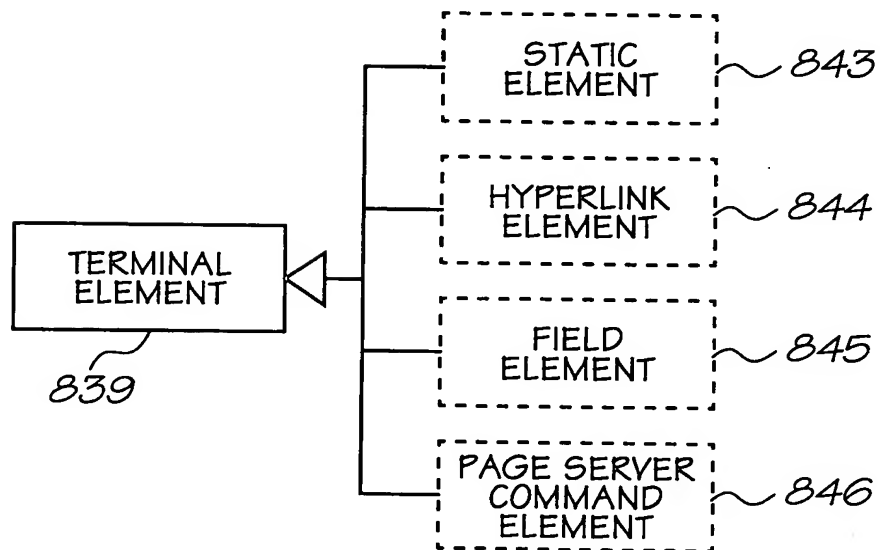


FIG. 27

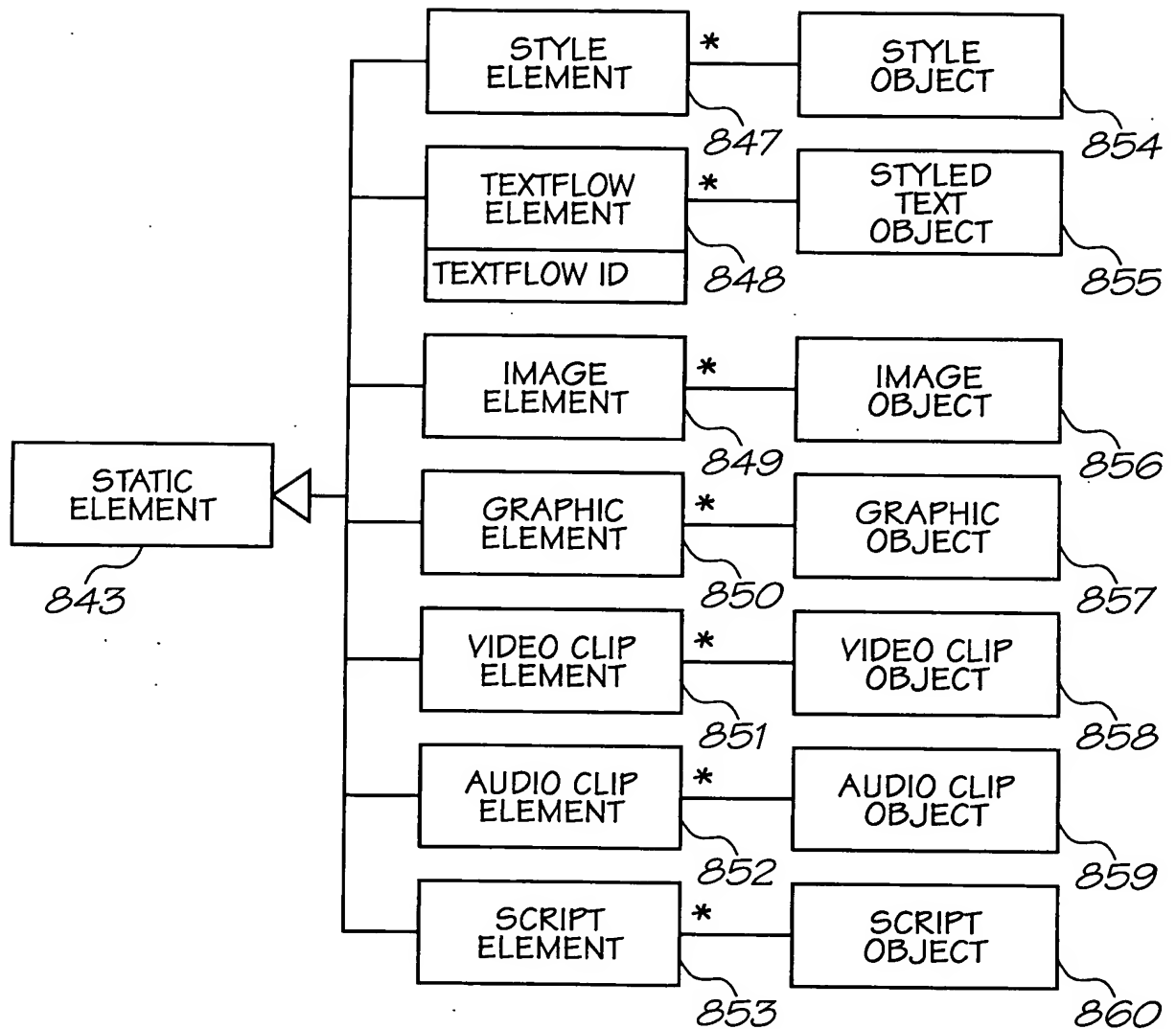


FIG. 28

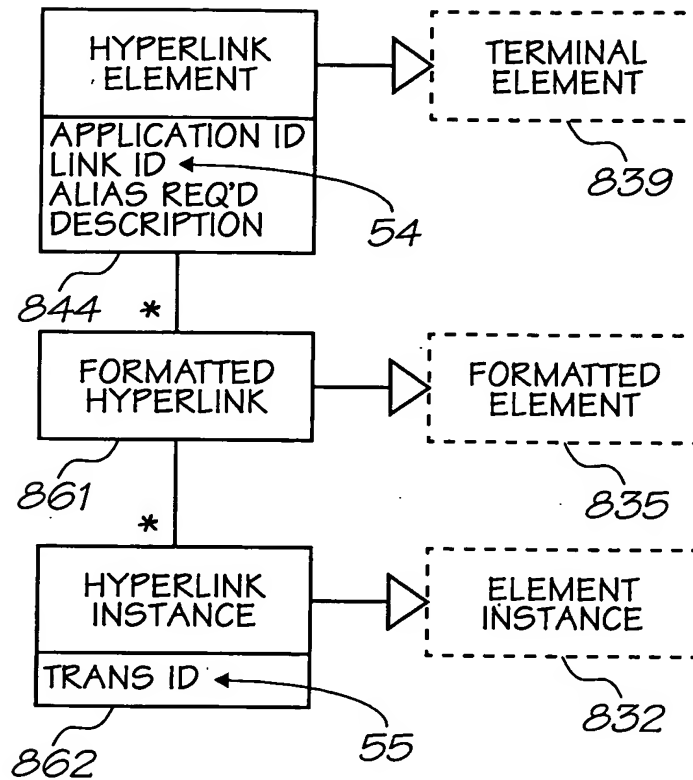


FIG. 29

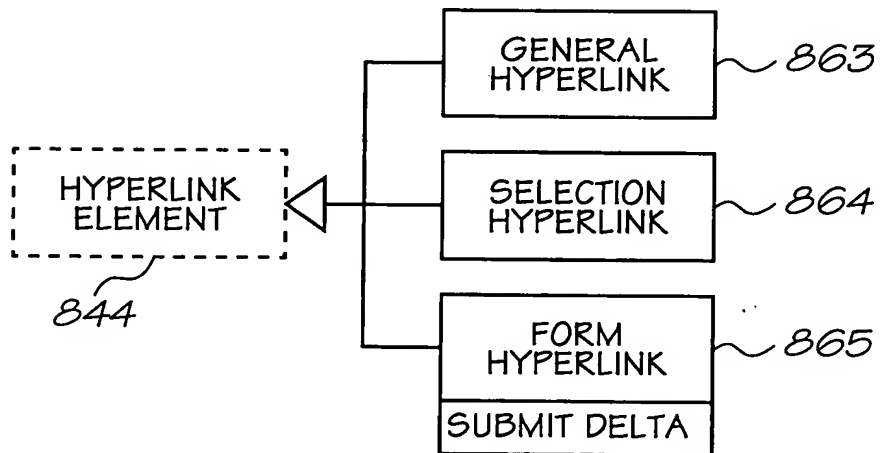


FIG. 30



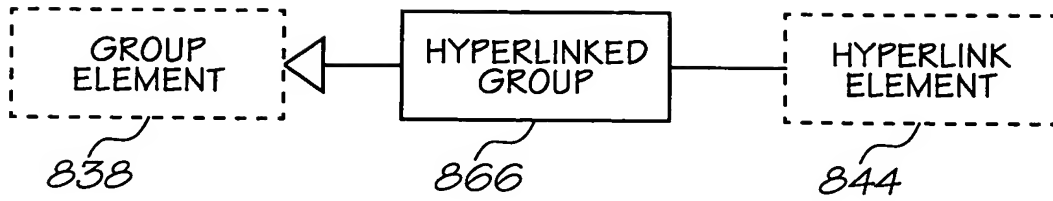


FIG. 31

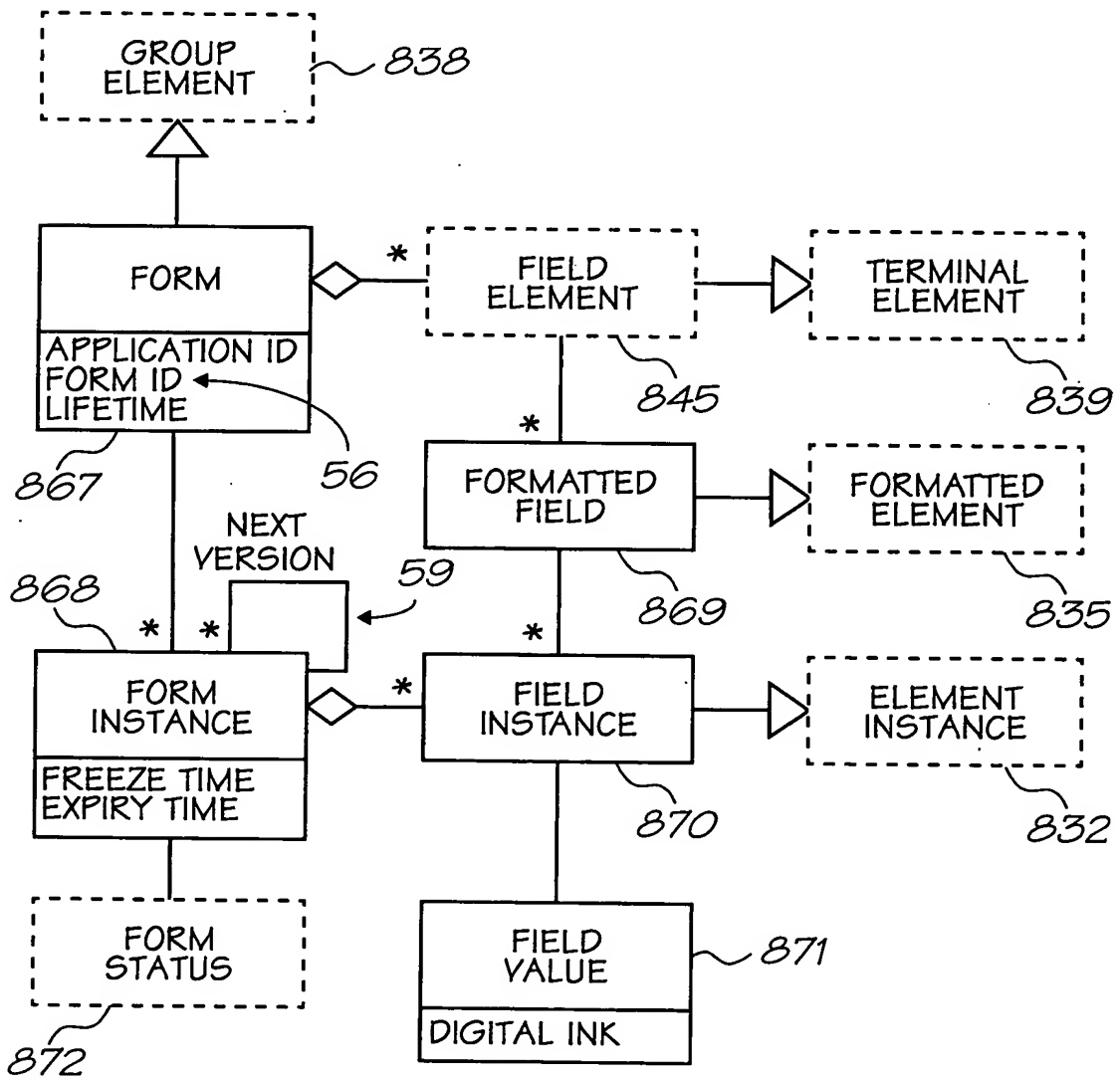


FIG. 32

30/157

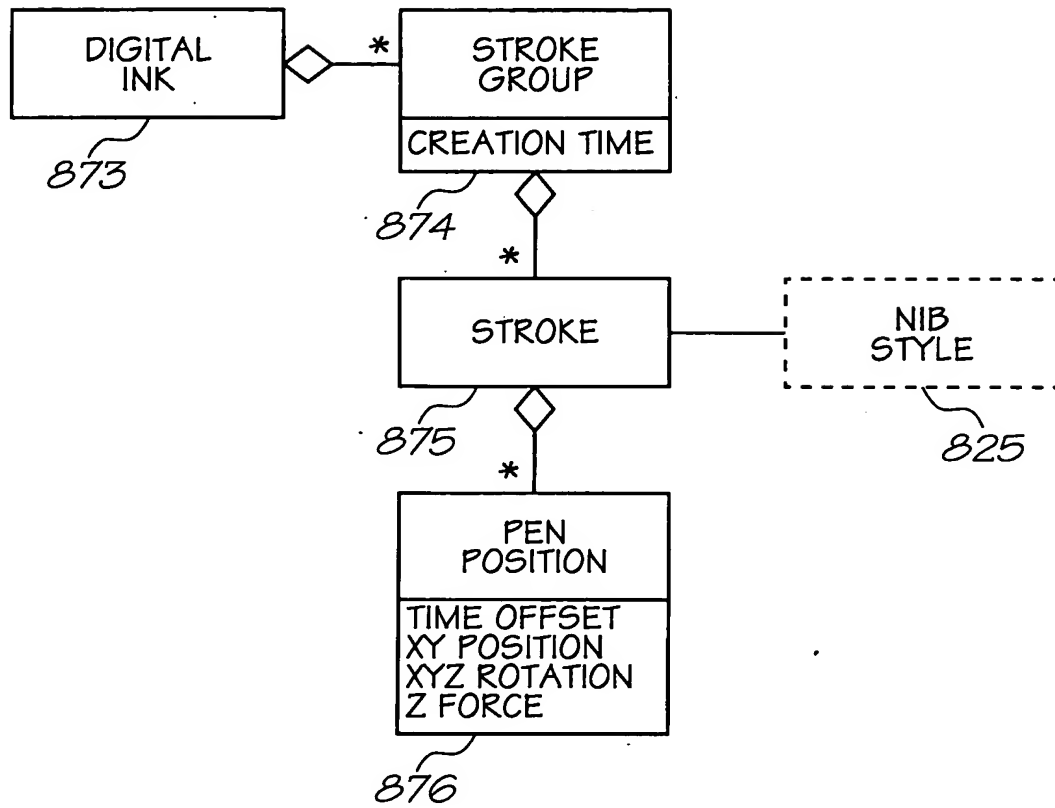


FIG. 33

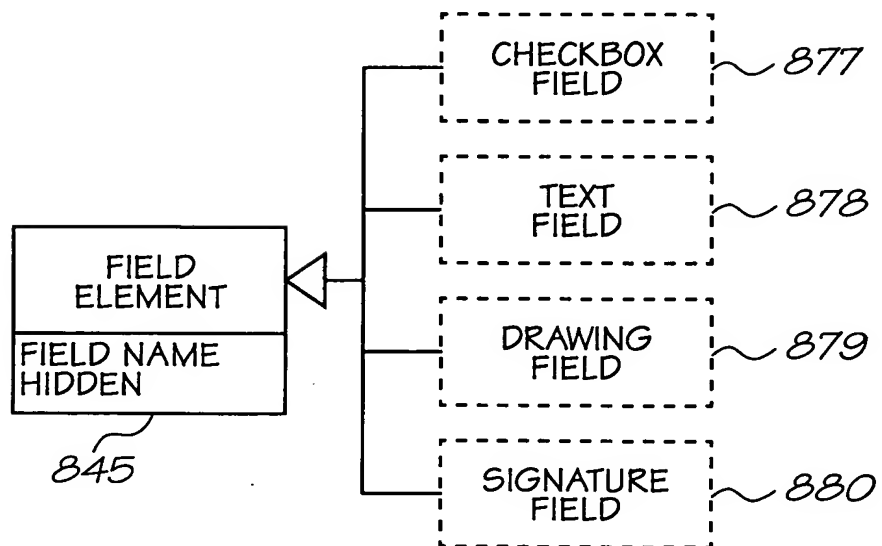


FIG. 34

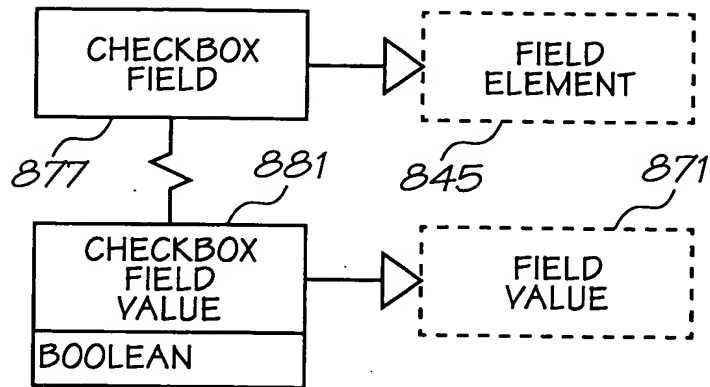


FIG. 35

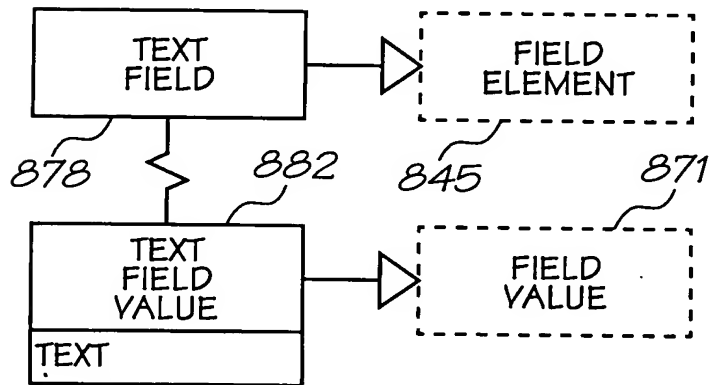


FIG. 36

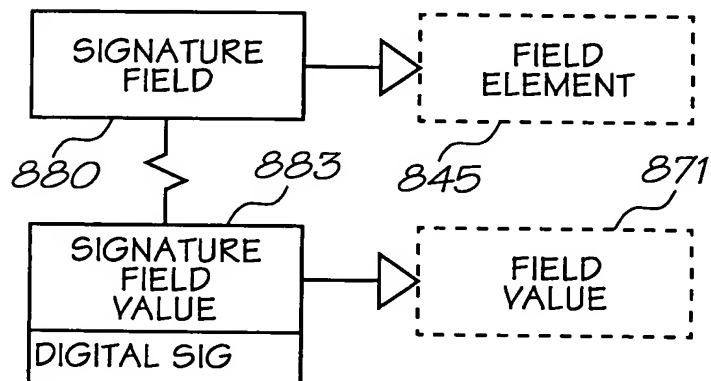


FIG. 37

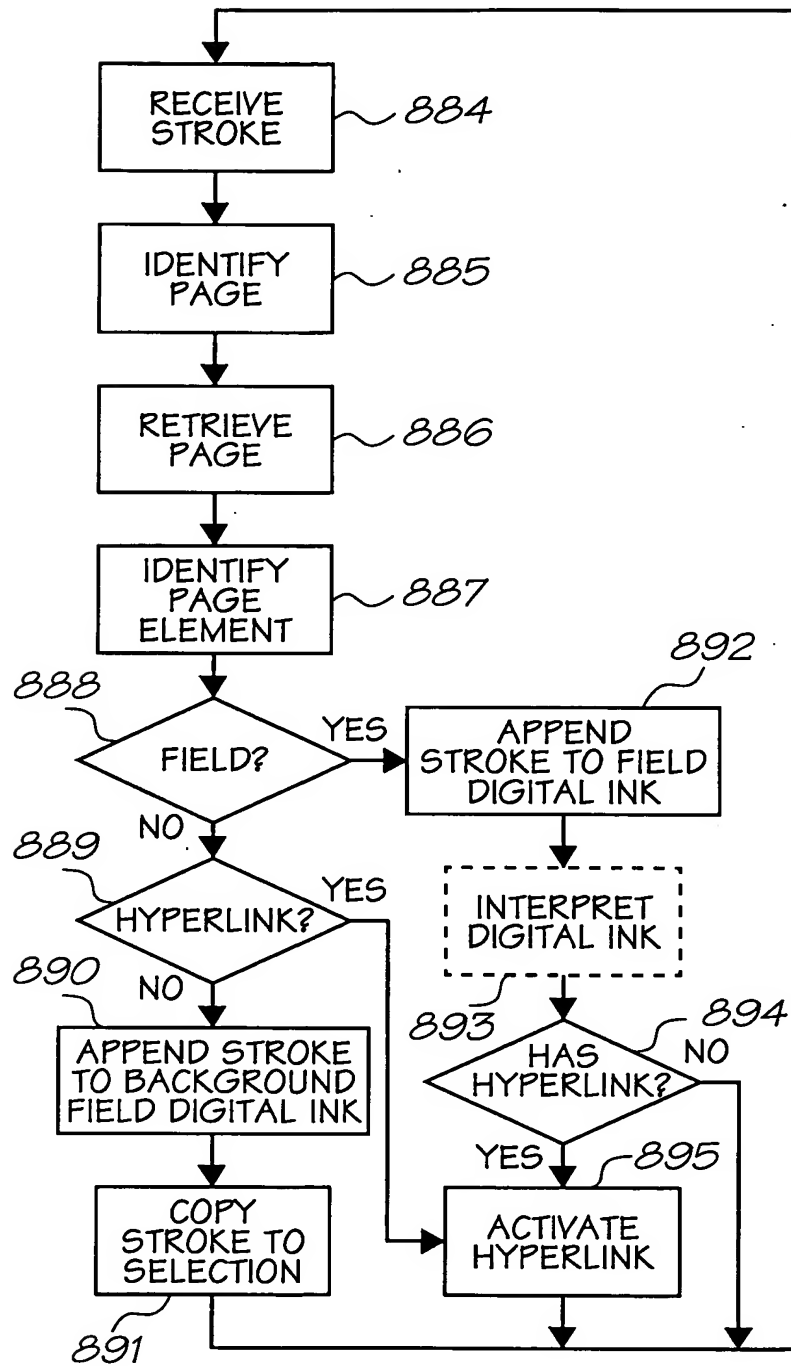


FIG. 38

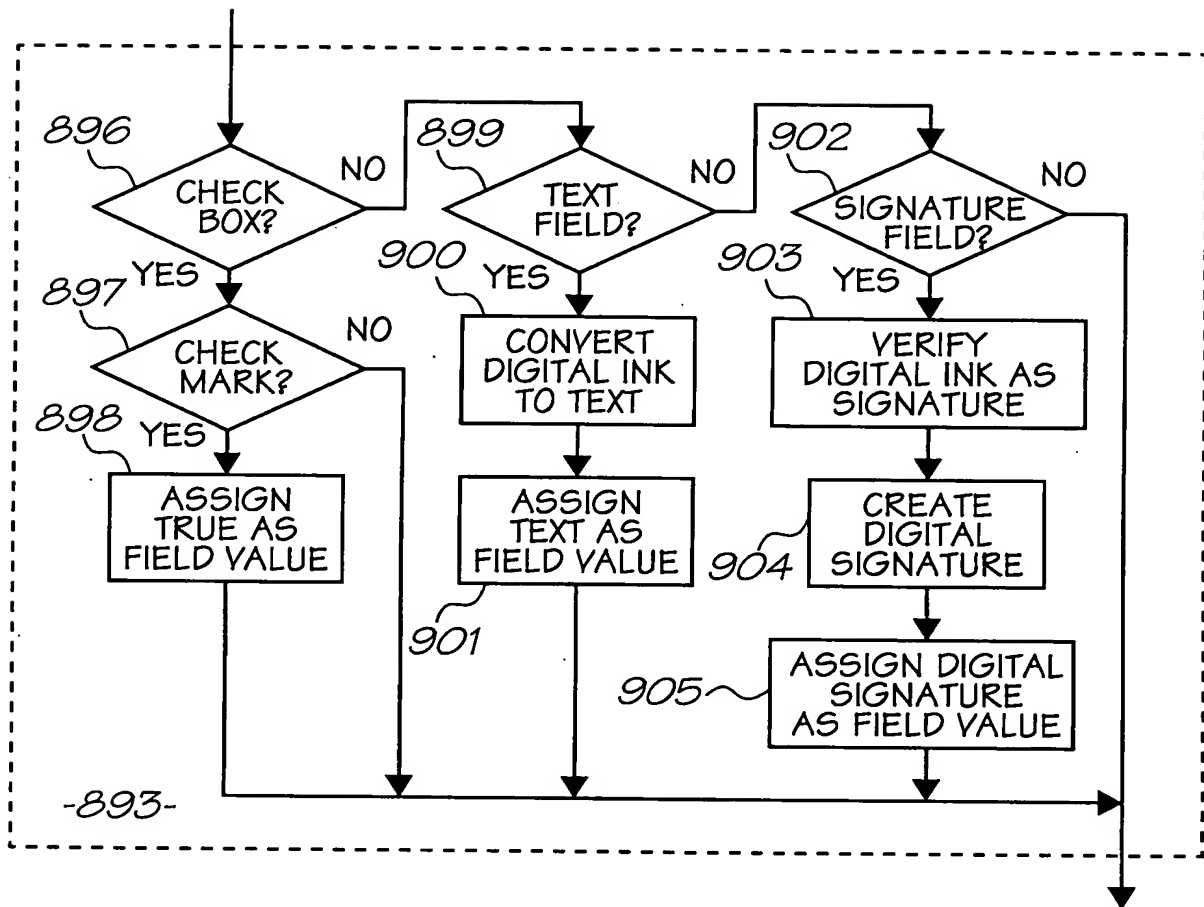


FIG. 38a

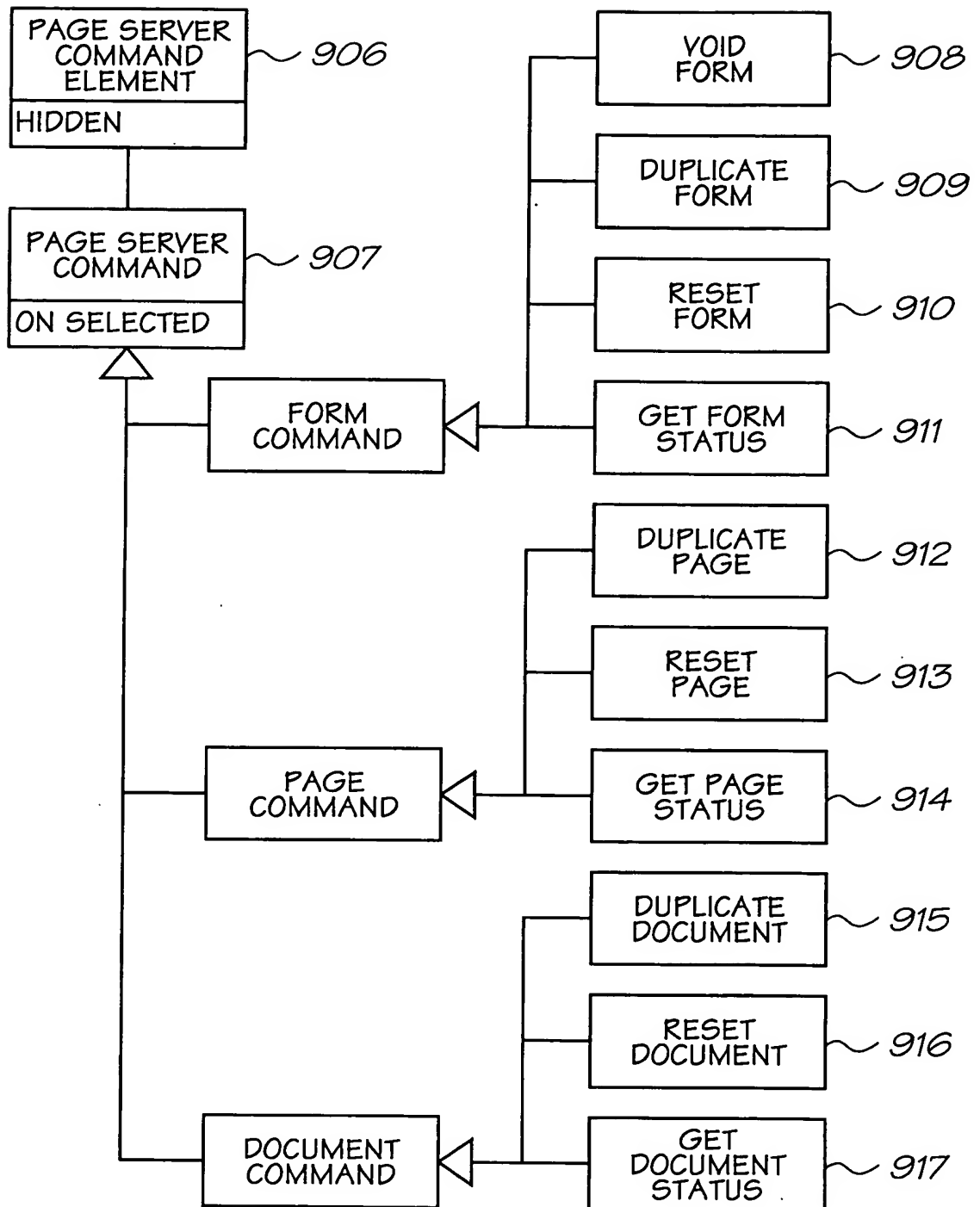


FIG. 39

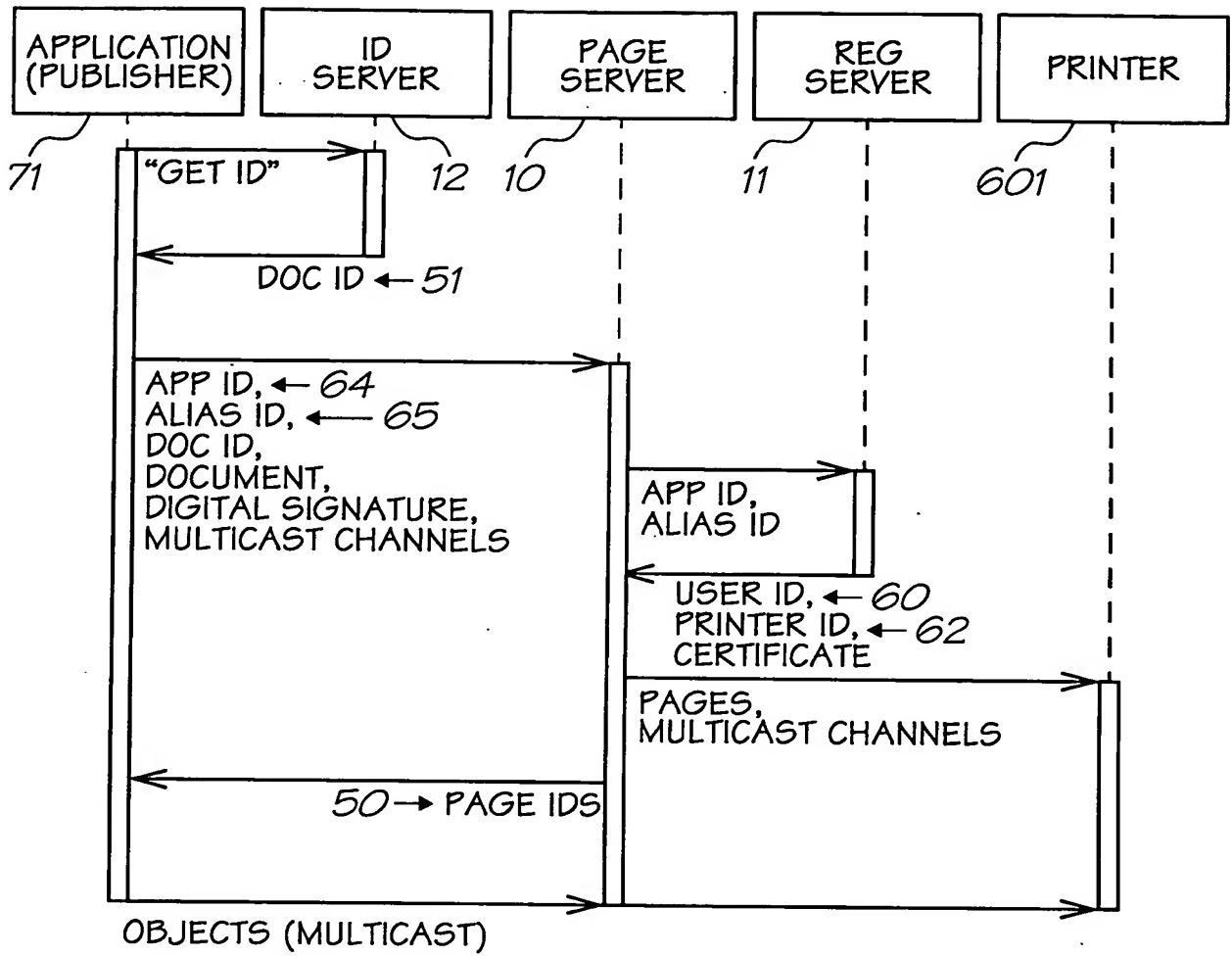


FIG. 40

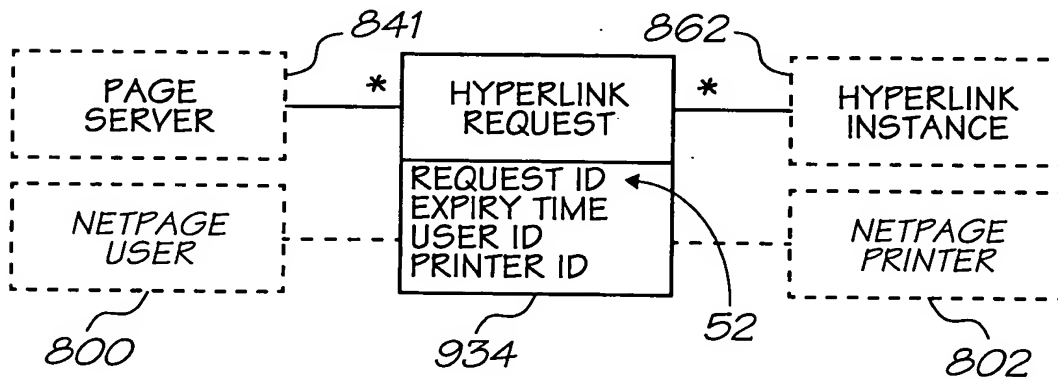


FIG. 41

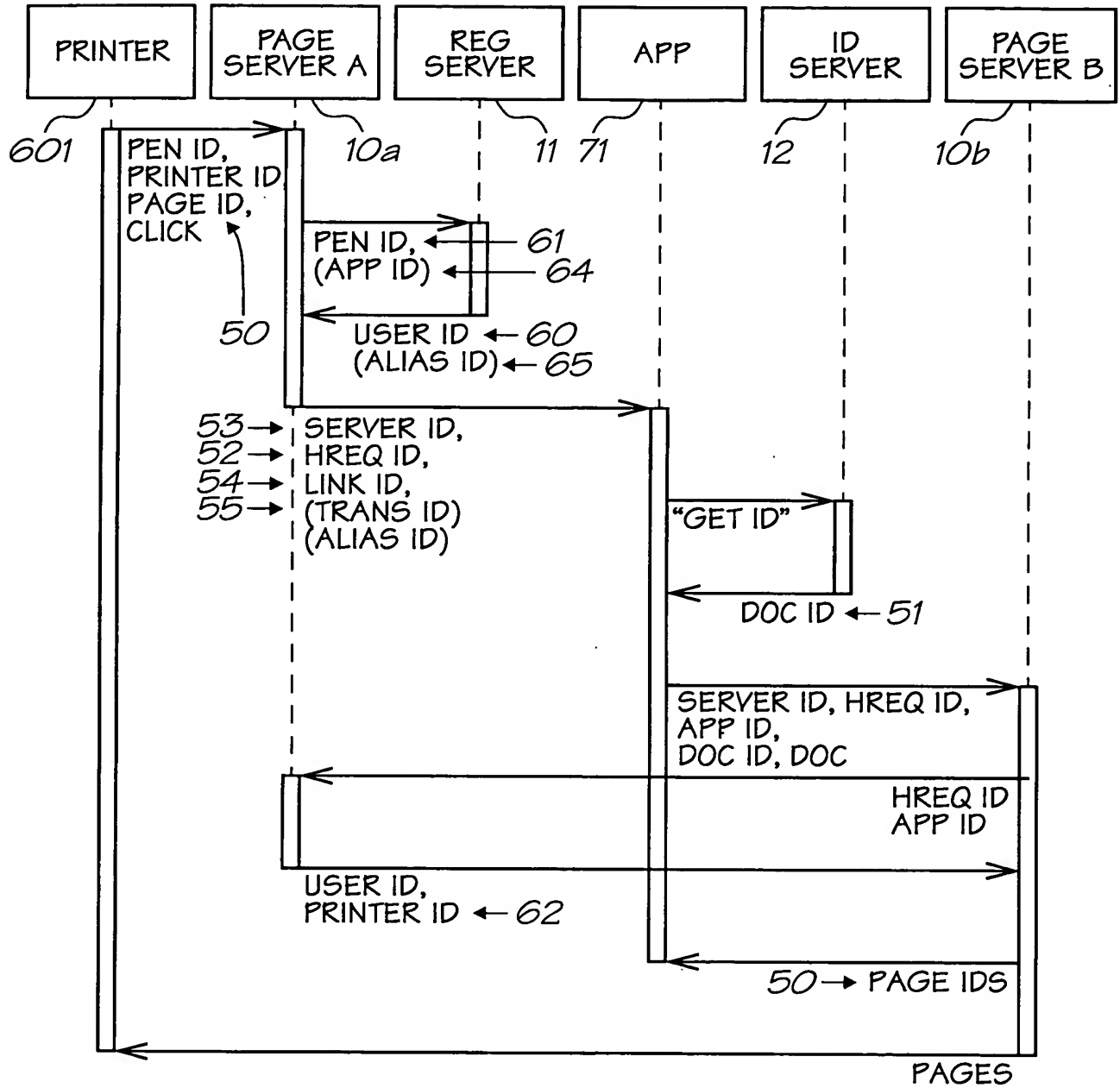


FIG. 42



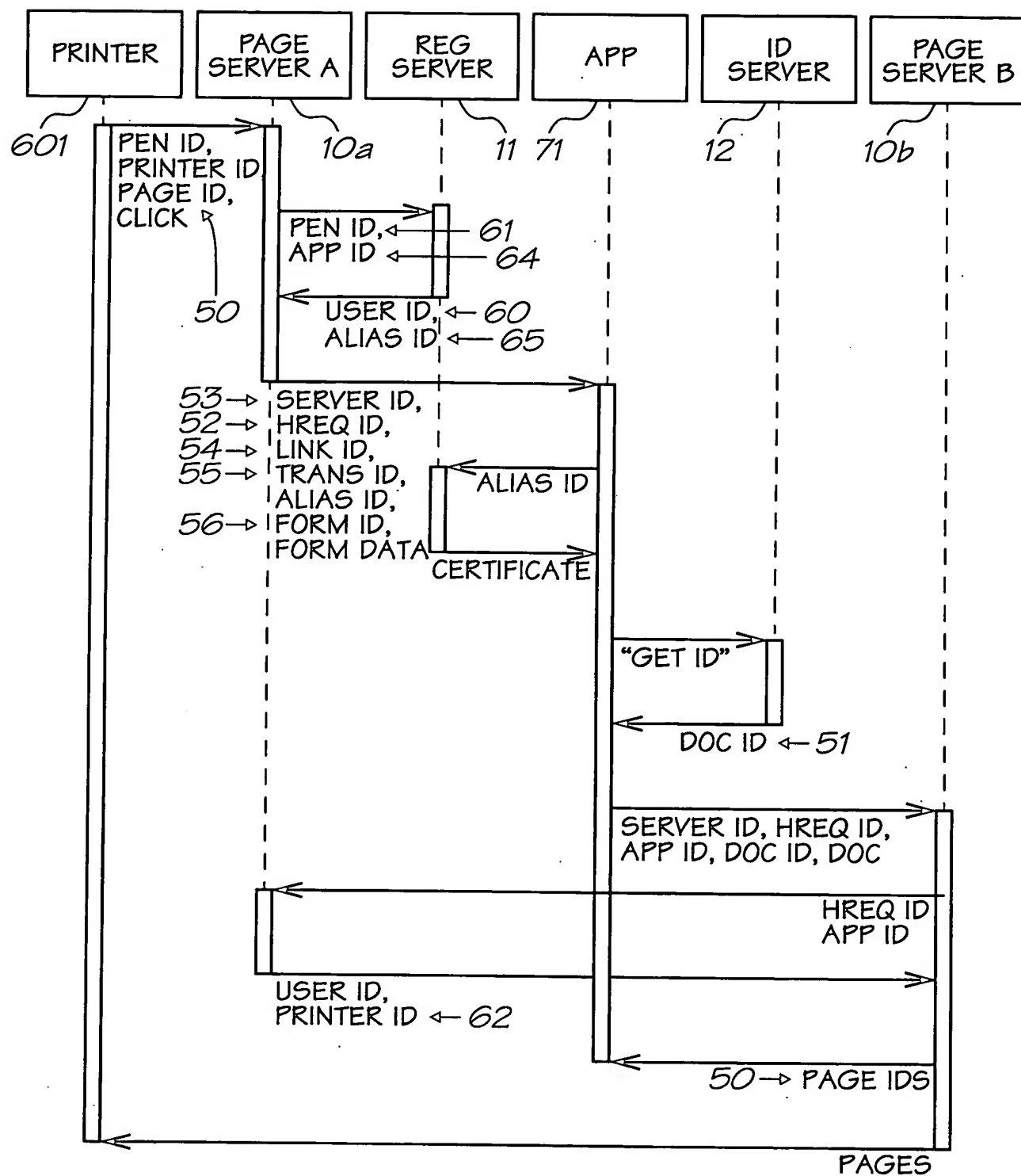


FIG. 43

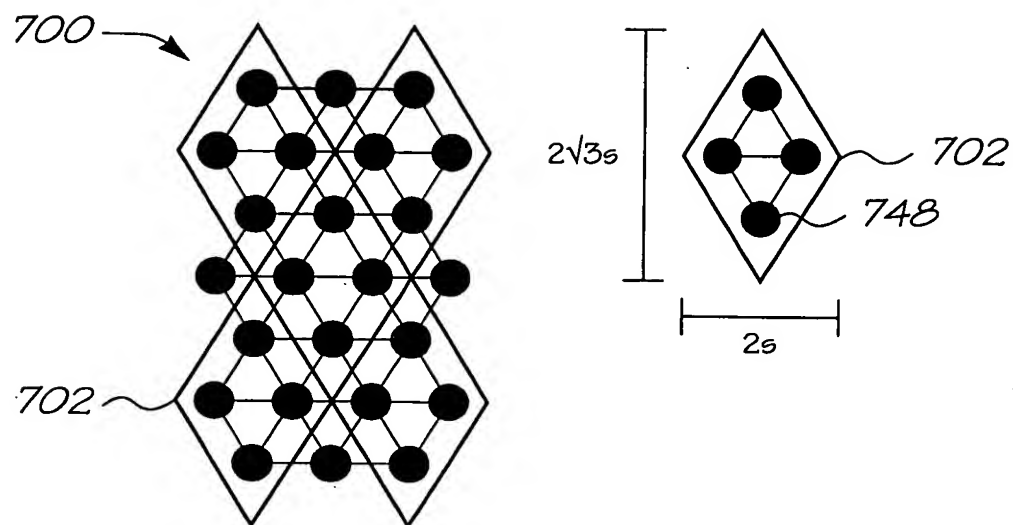


FIG. 44

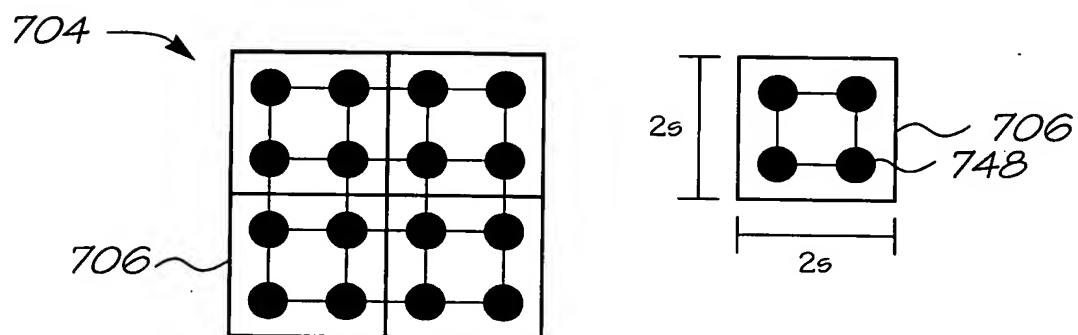


FIG. 45

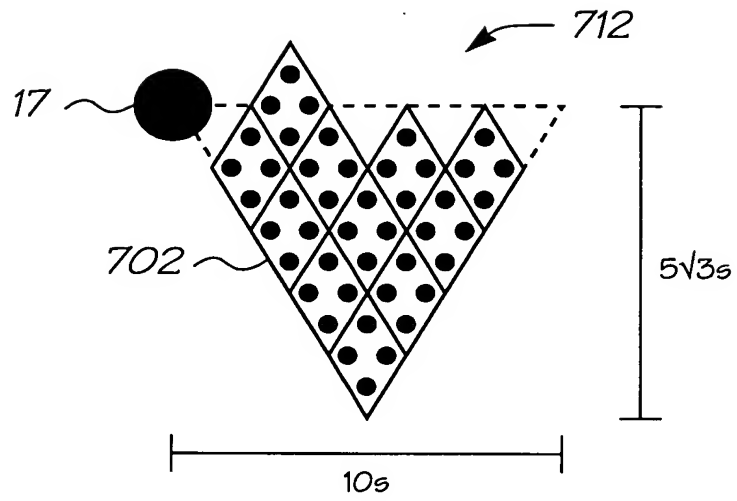


FIG. 46

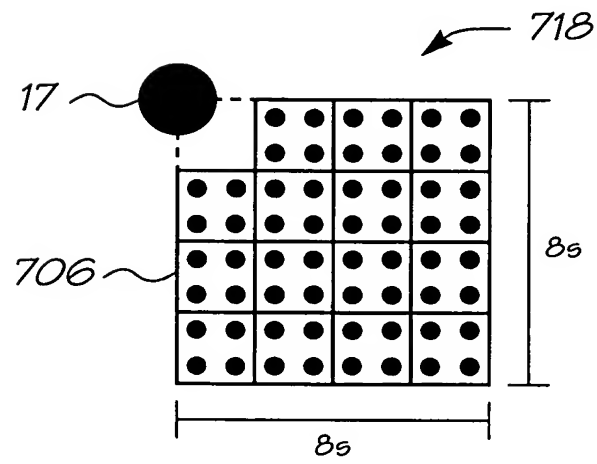


FIG. 47

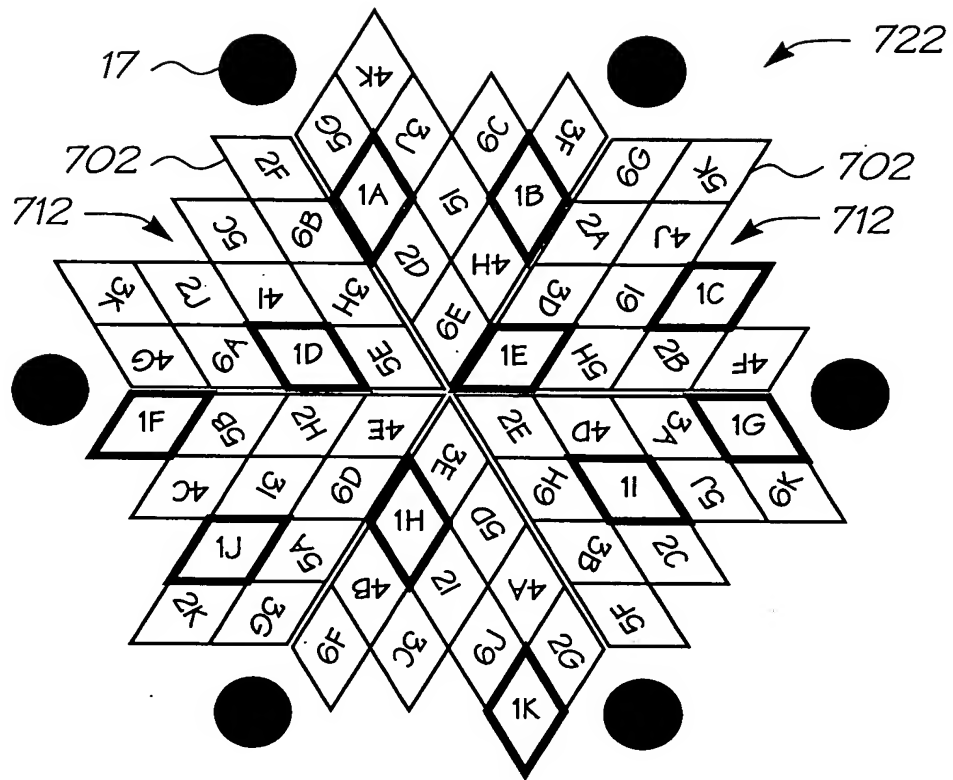


FIG. 48

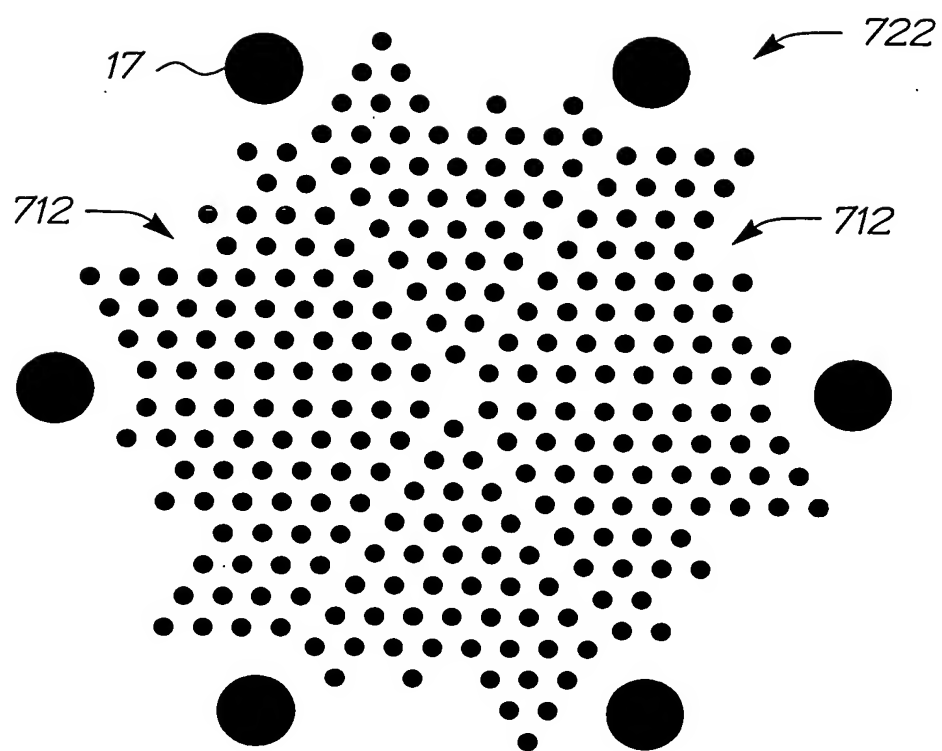


FIG. 49

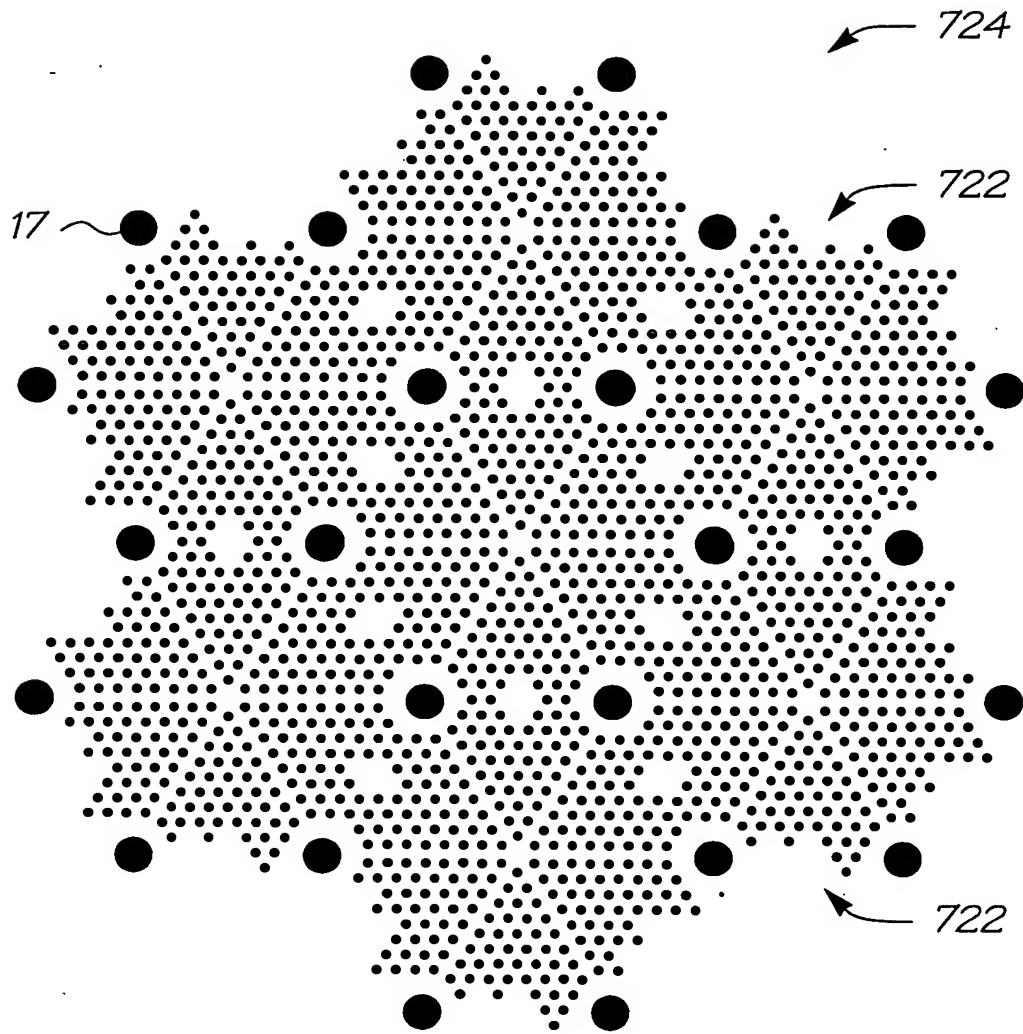


FIG. 50

43/157

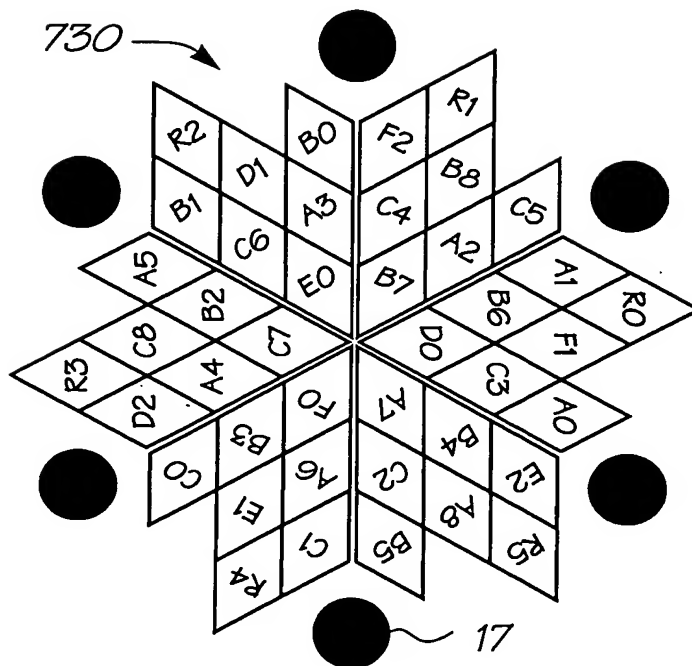


FIG. 51

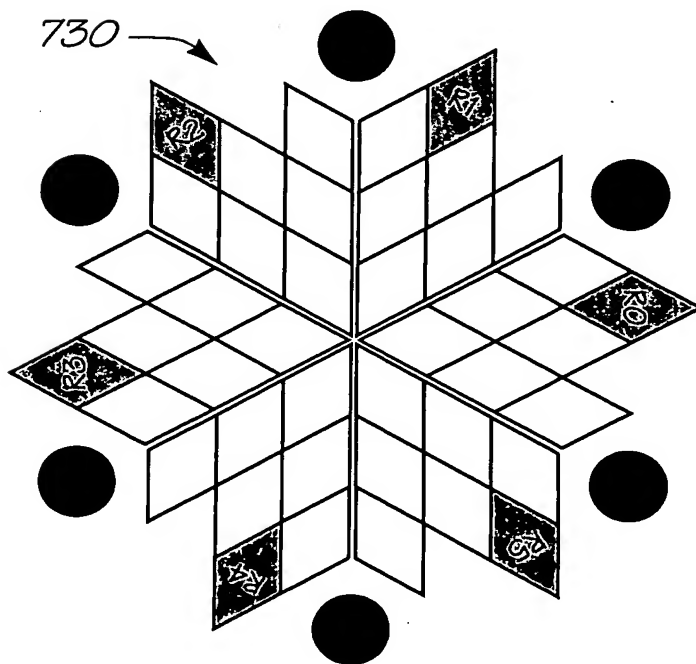


FIG. 52

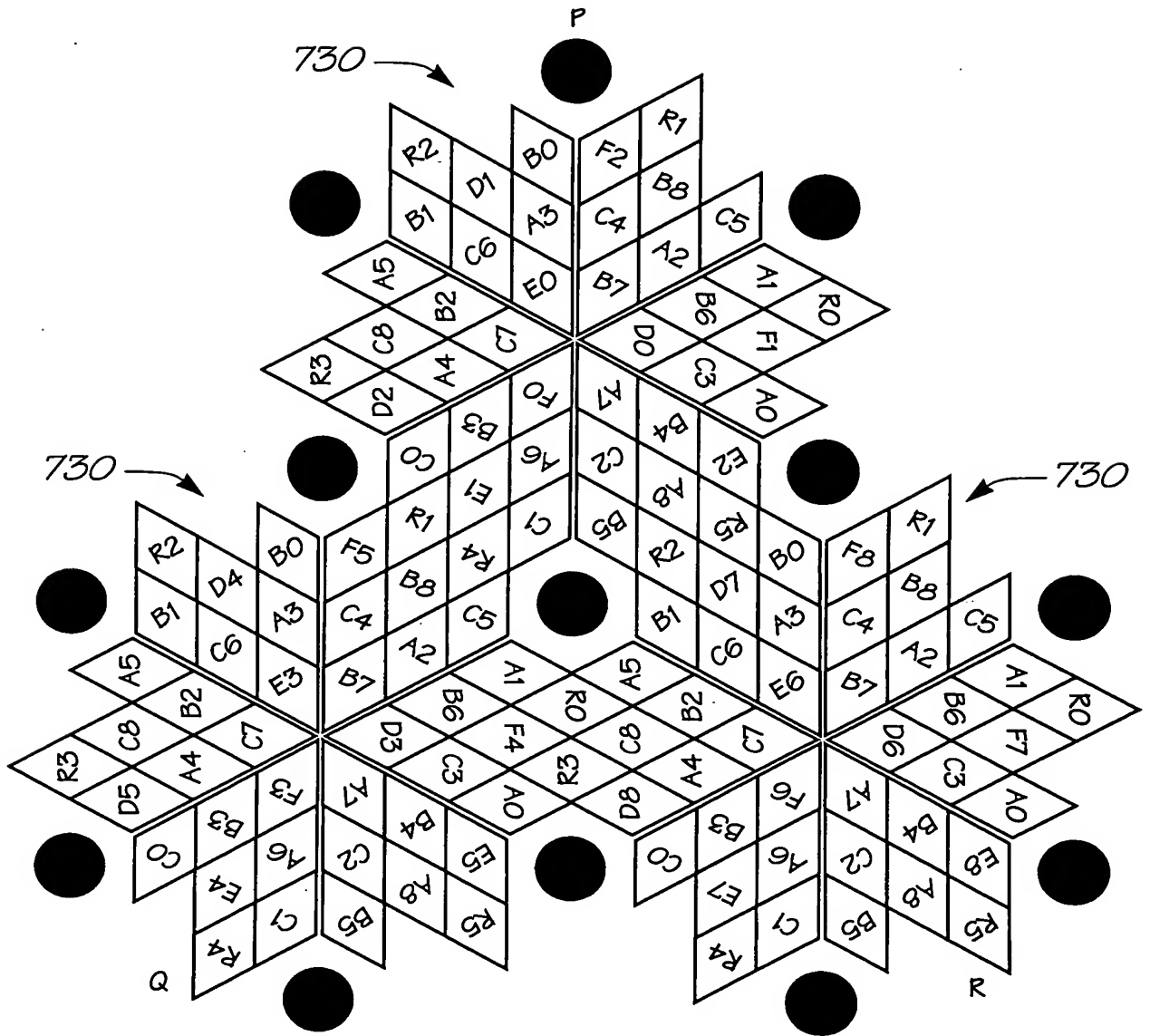


FIG. 53



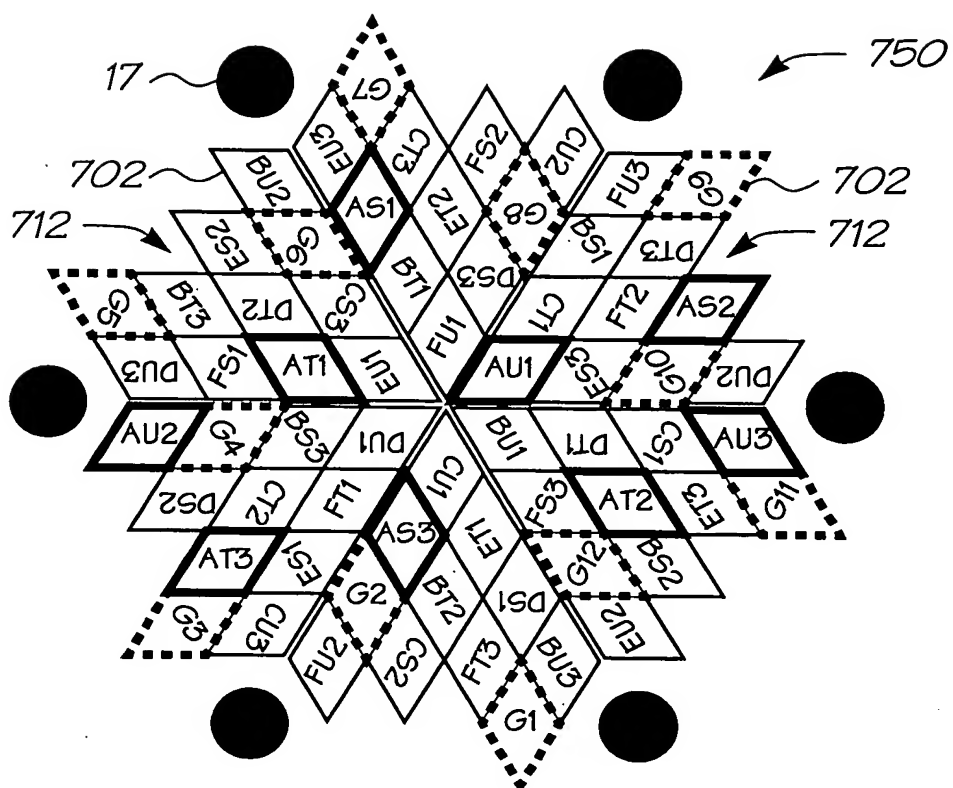


FIG. 54

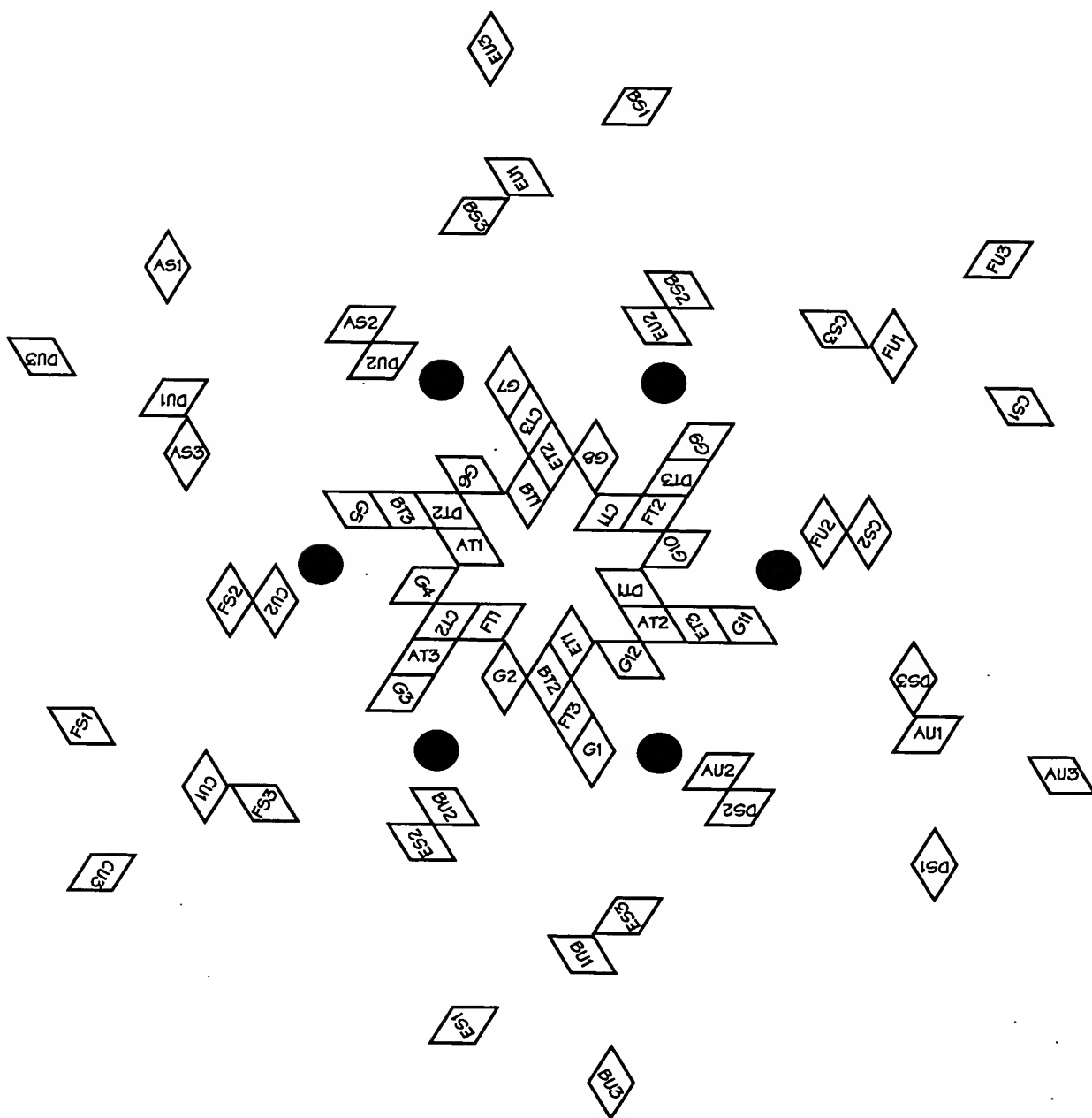


FIG. 55

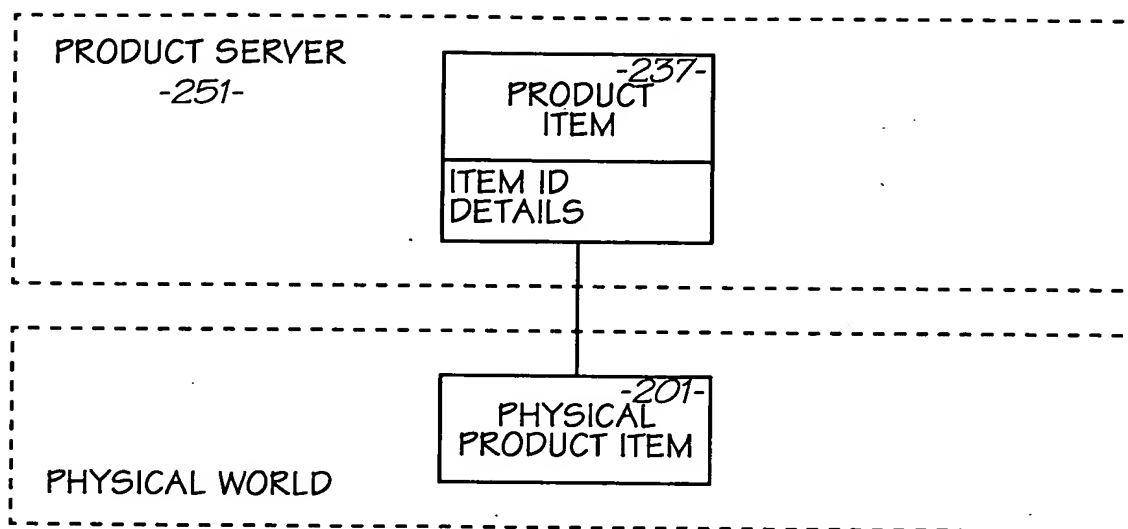


FIG. 56

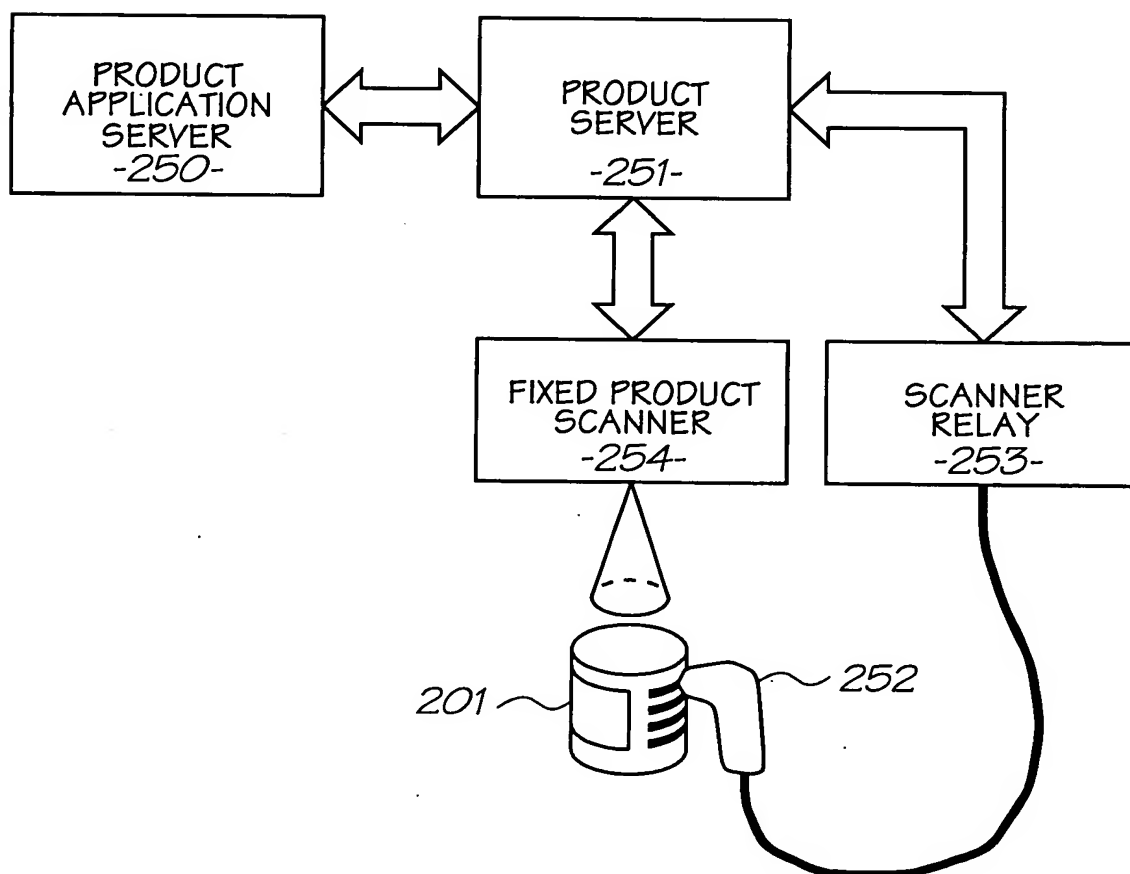


FIG. 57

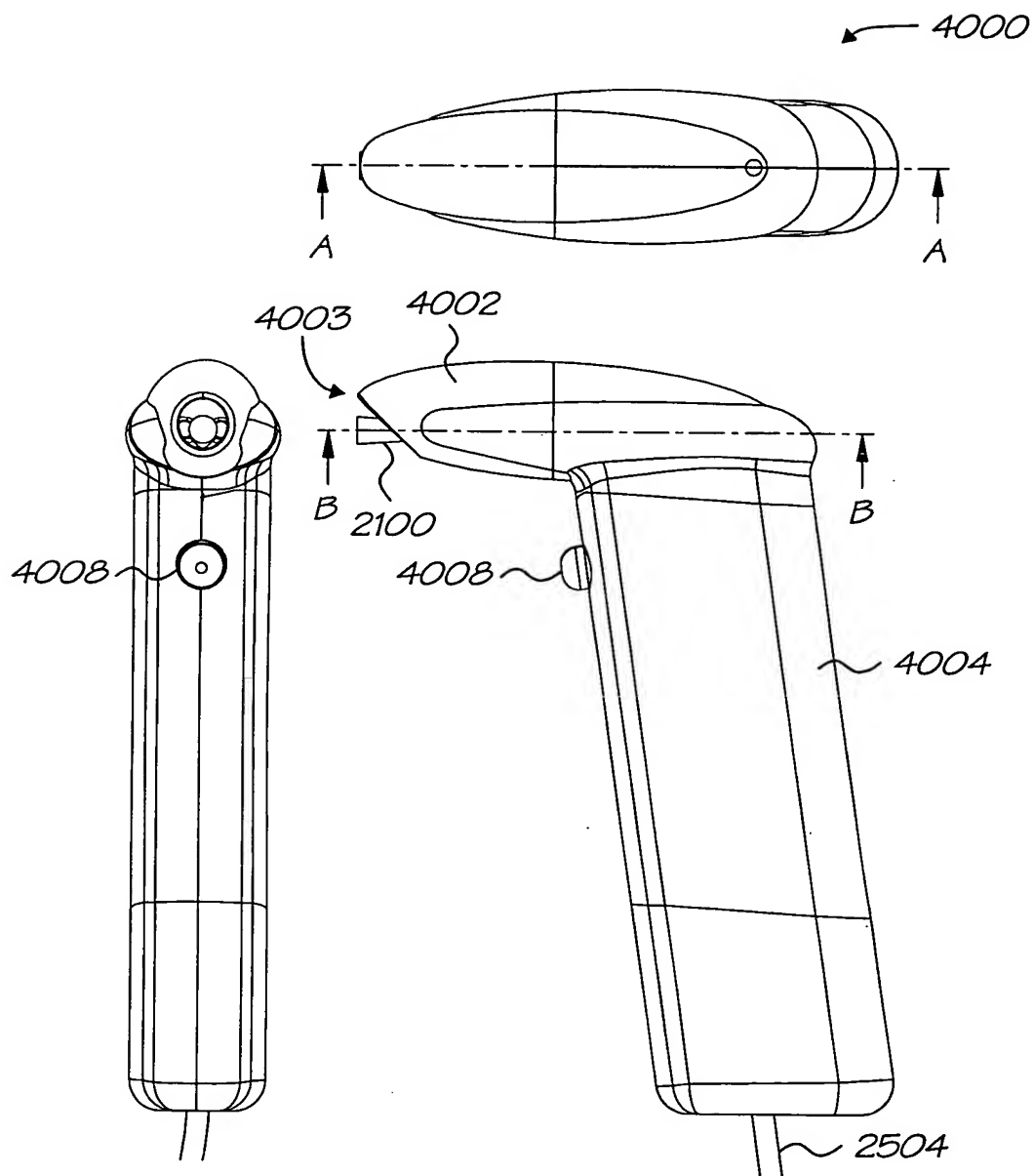


FIG. 58

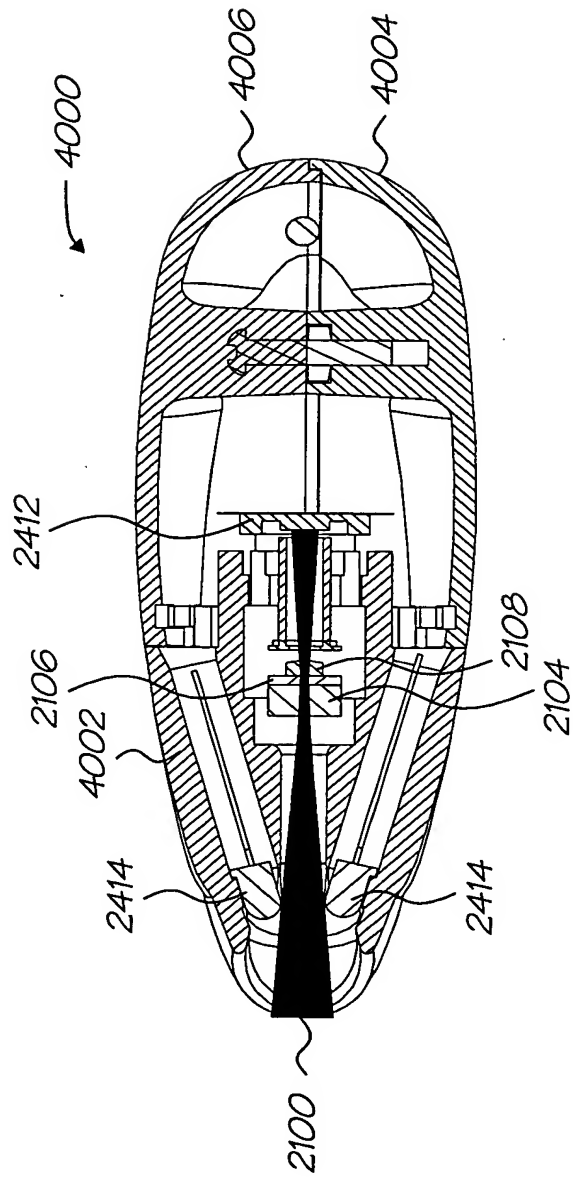


FIG. 59

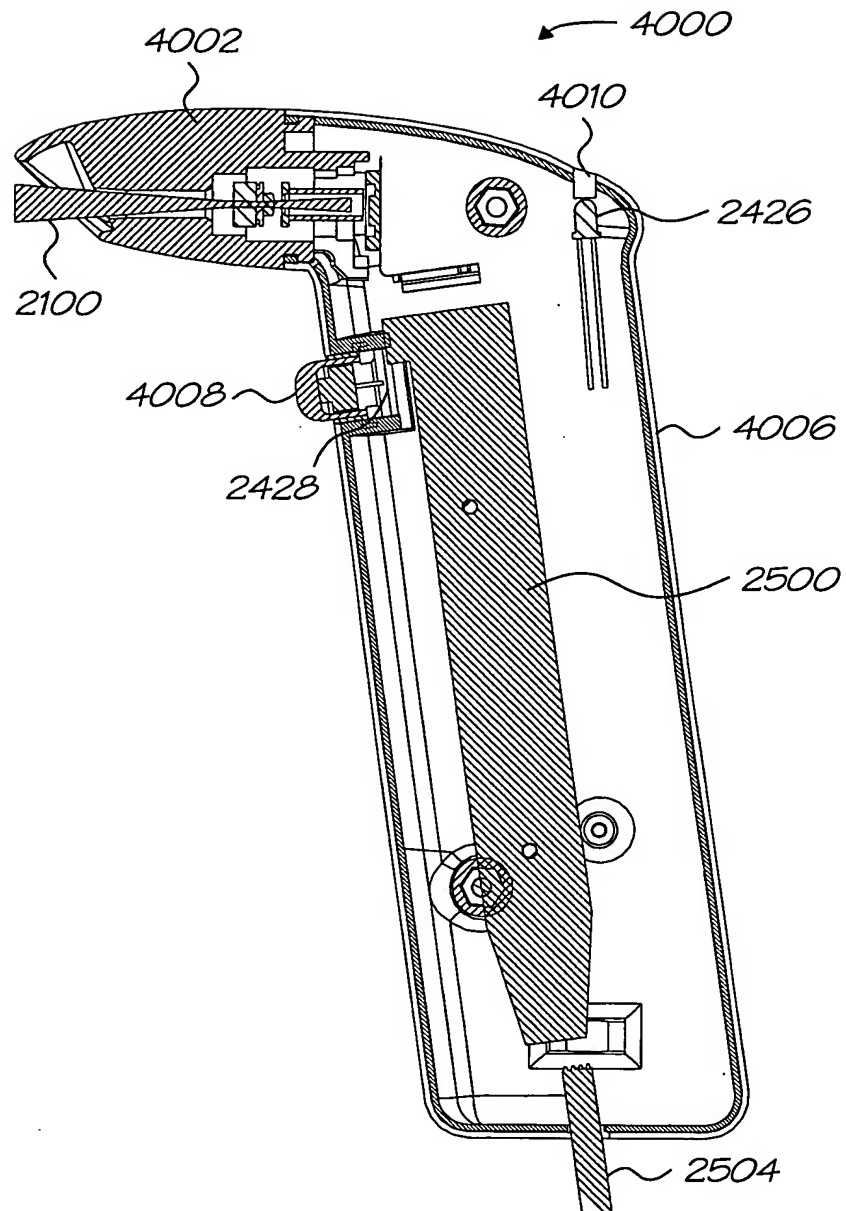


FIG. 60

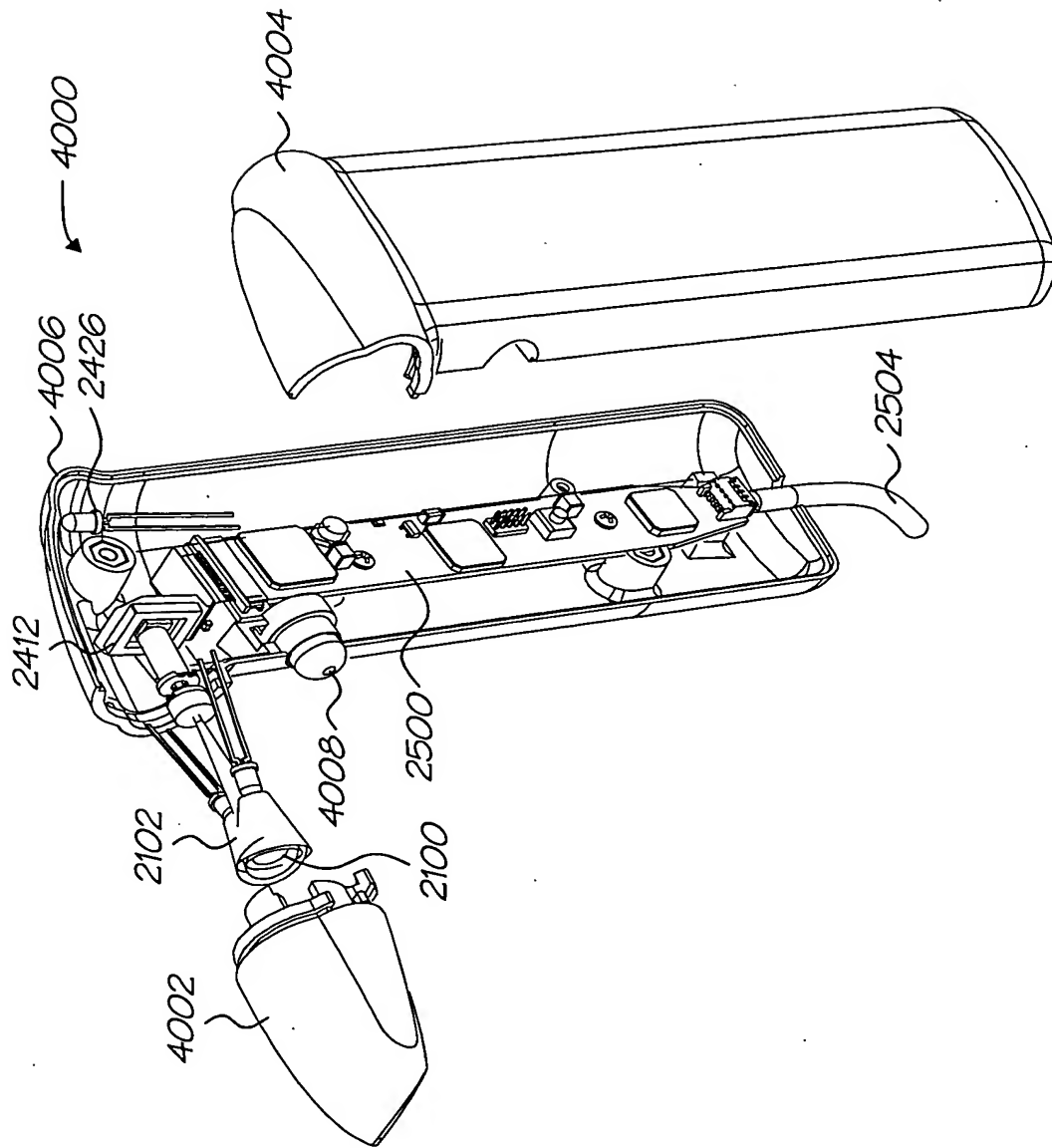


FIG. 61

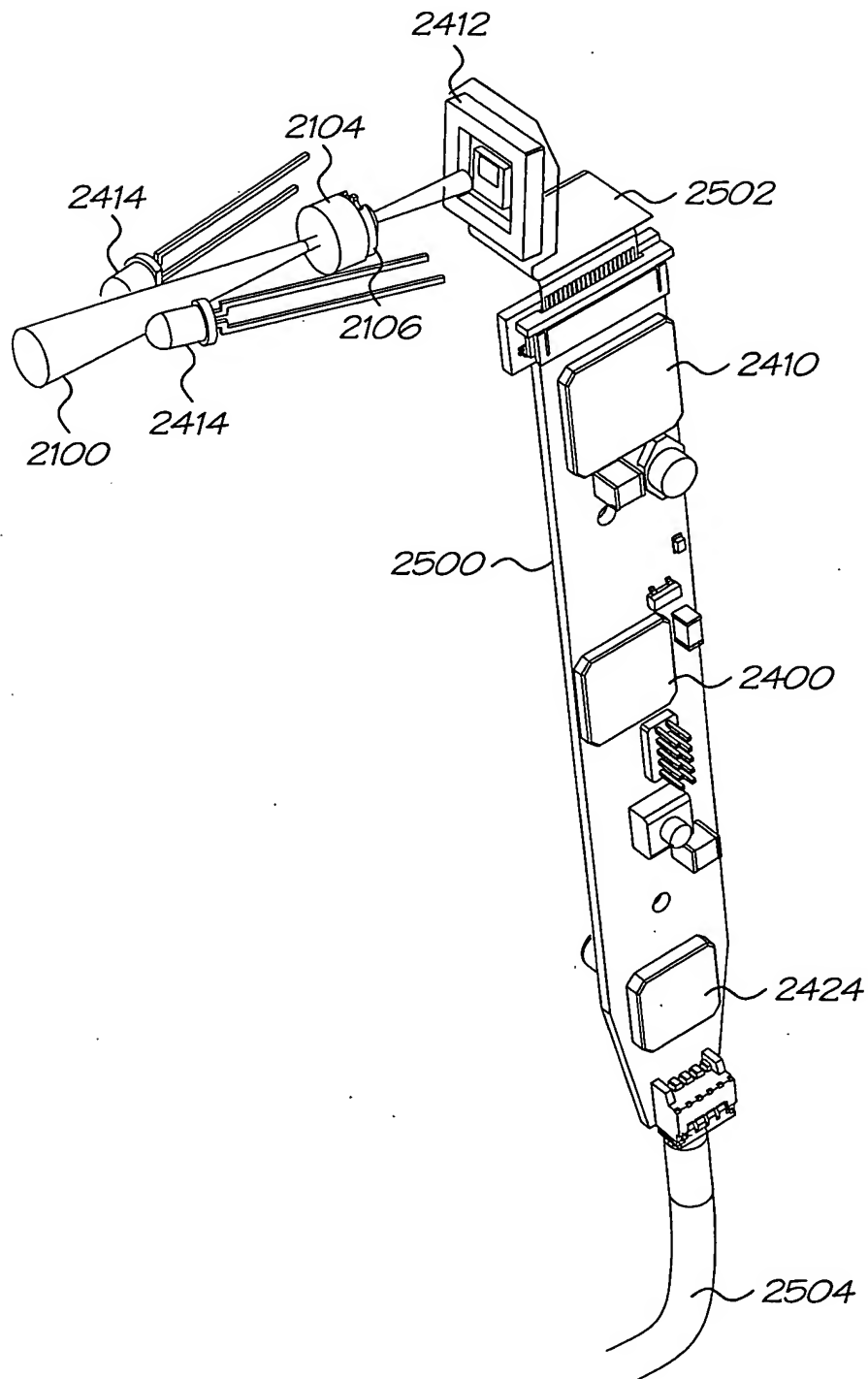


FIG. 62



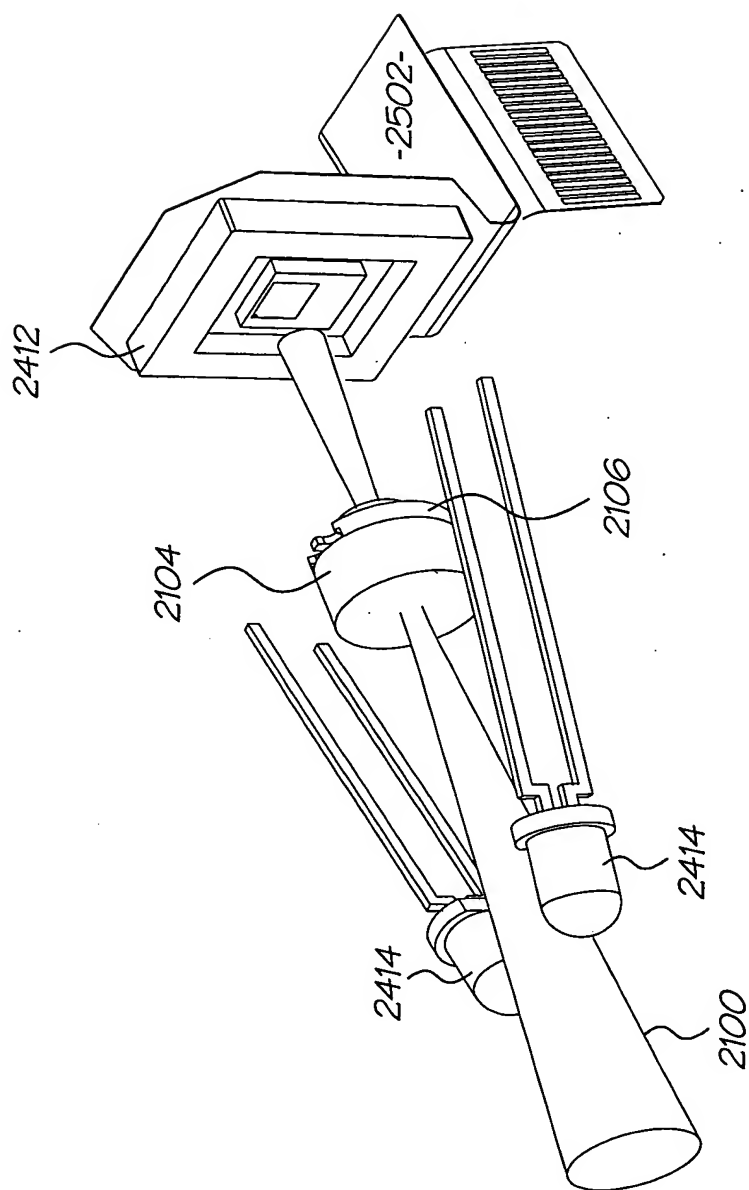


FIG. 63

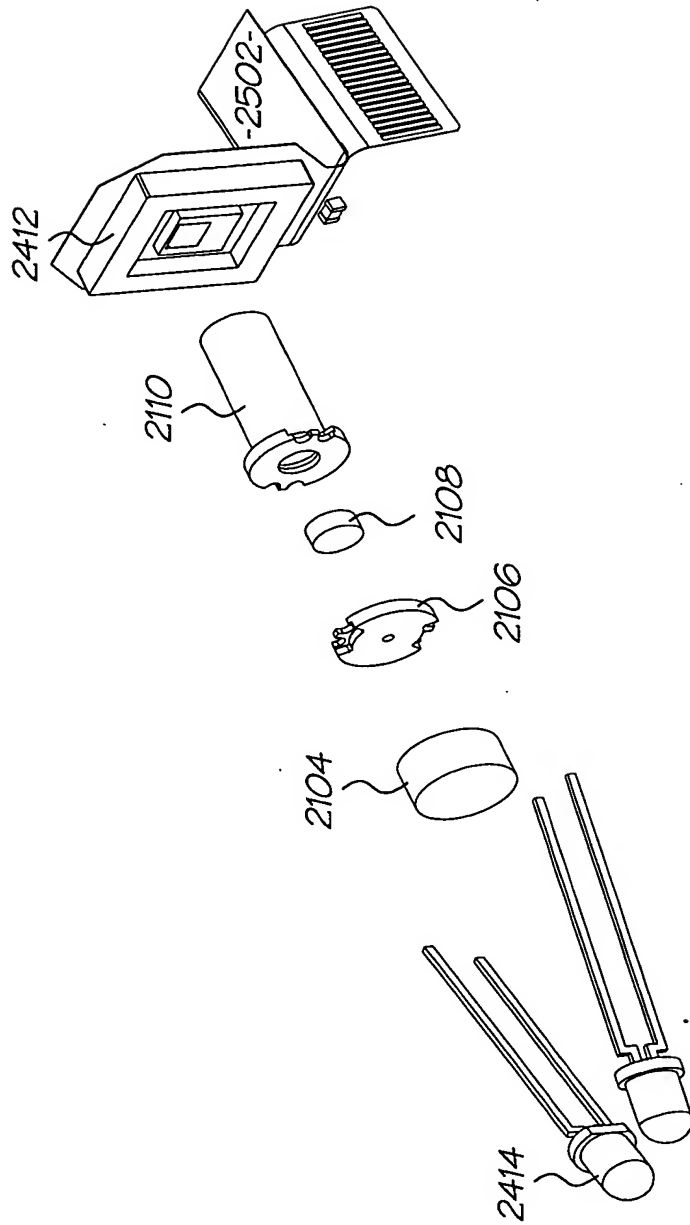


FIG. 64

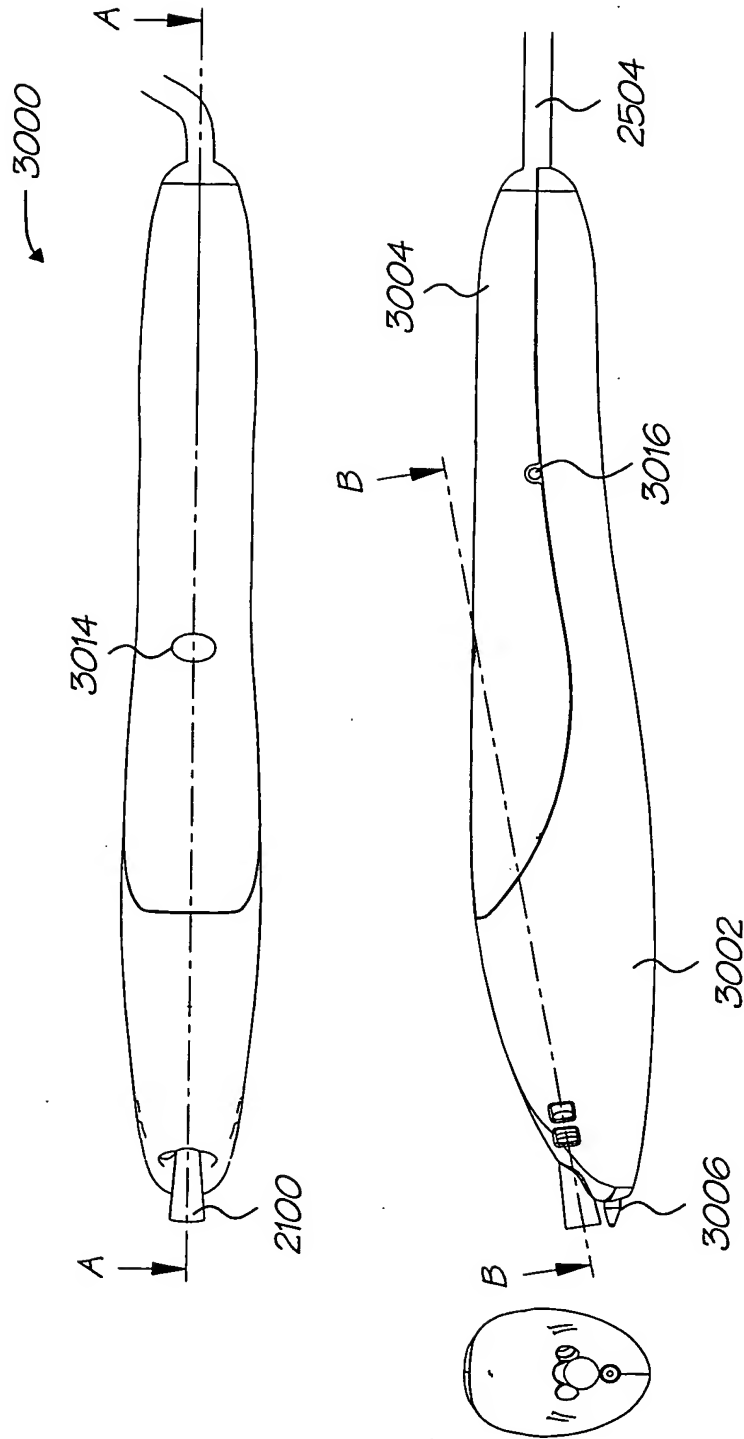


FIG. 65

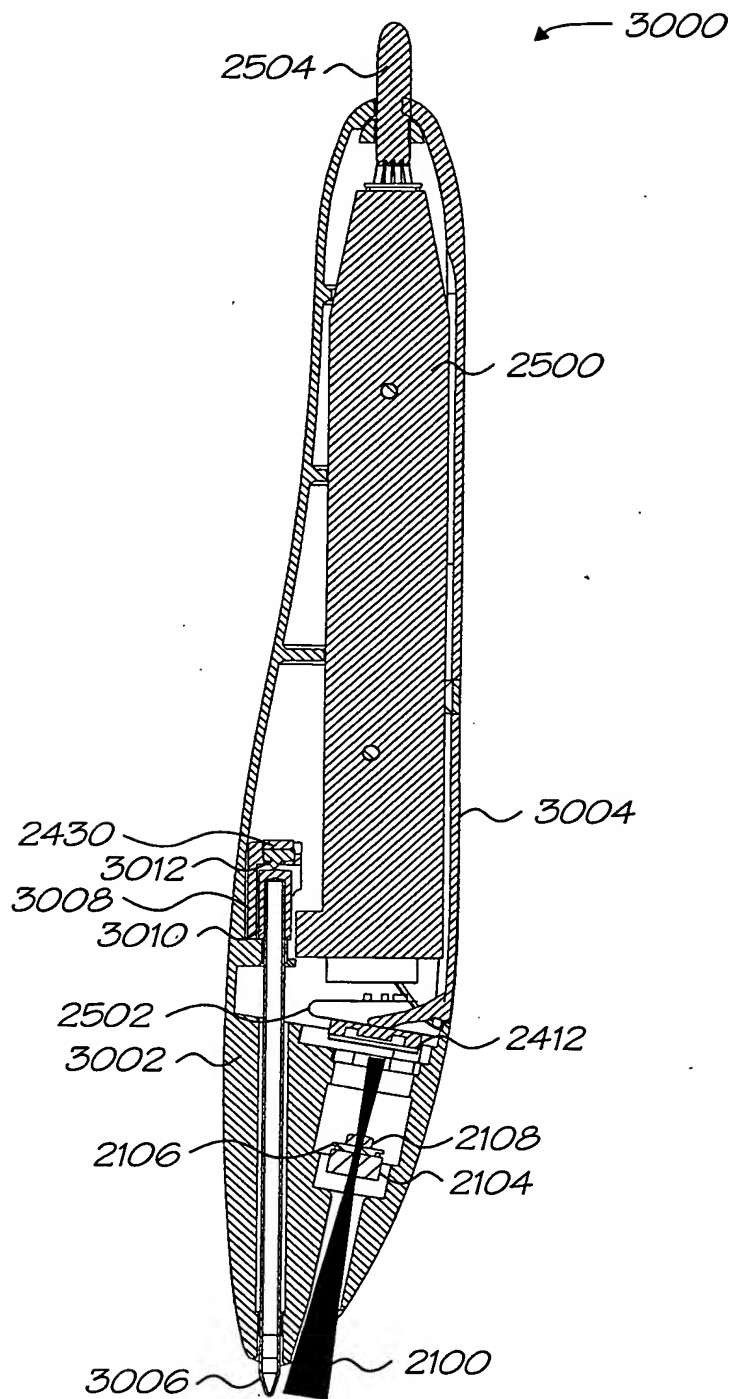


FIG. 66

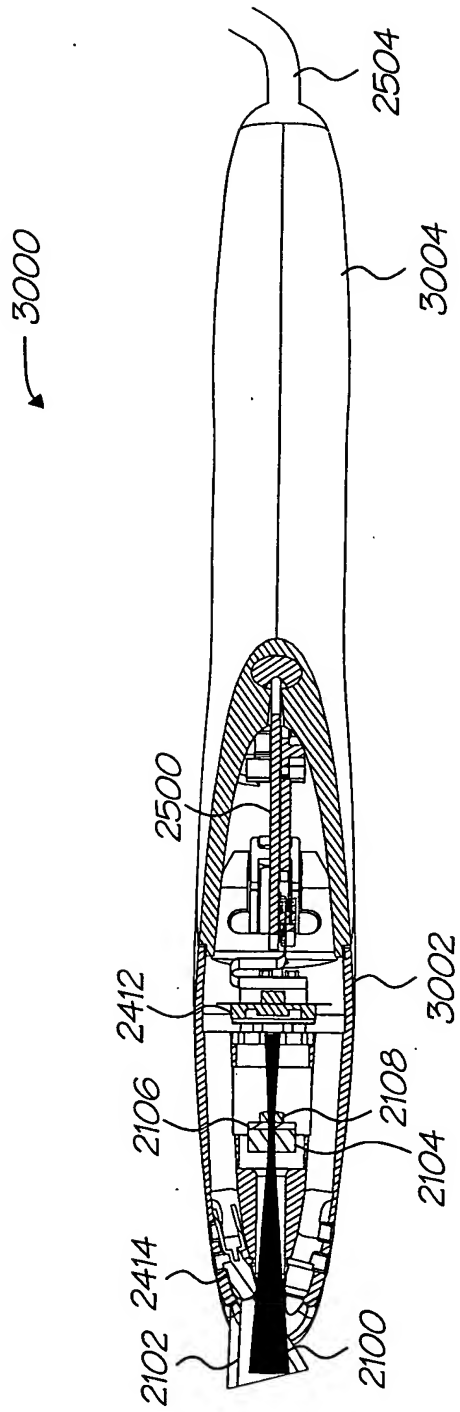


FIG. 67



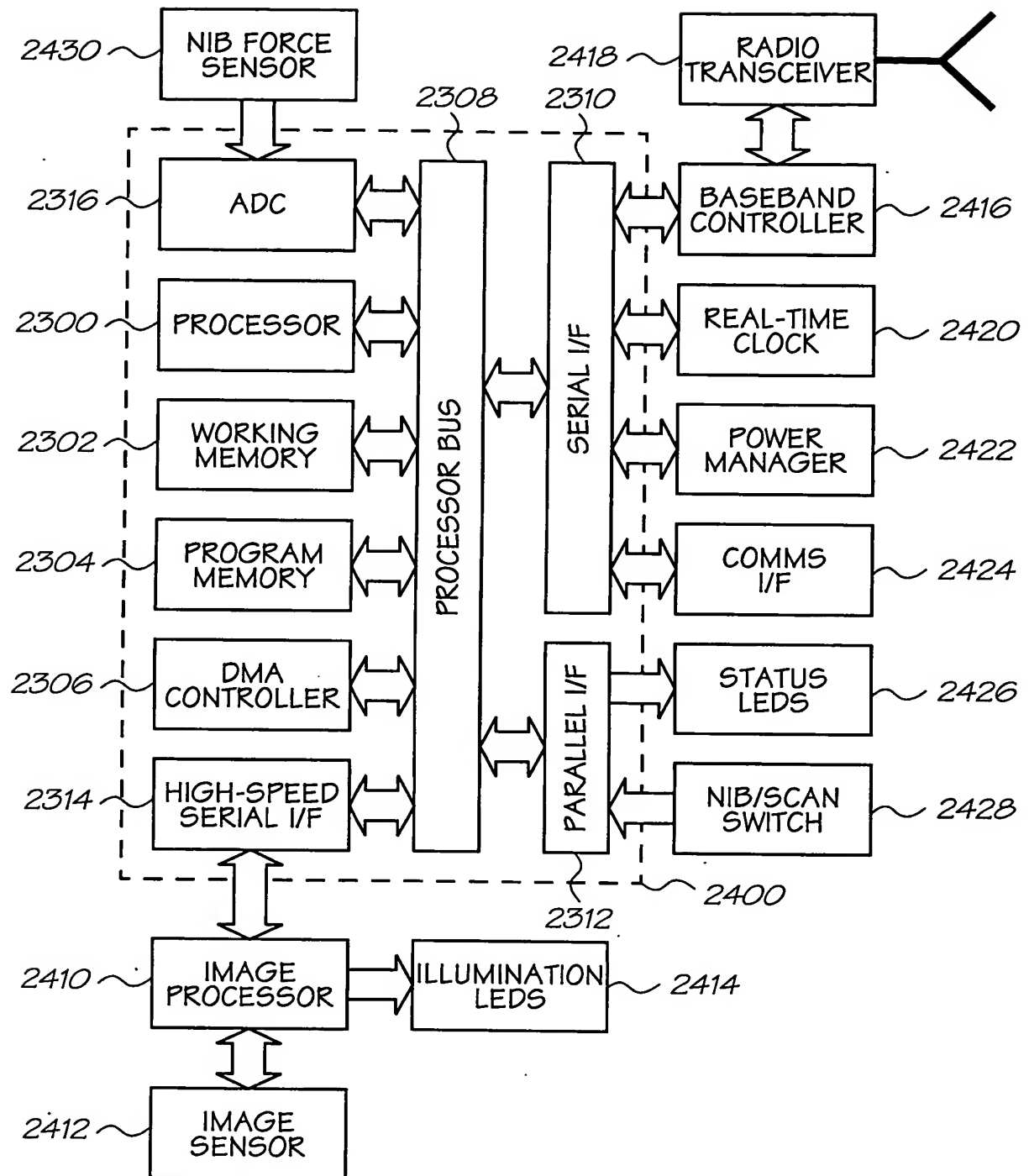


FIG. 69

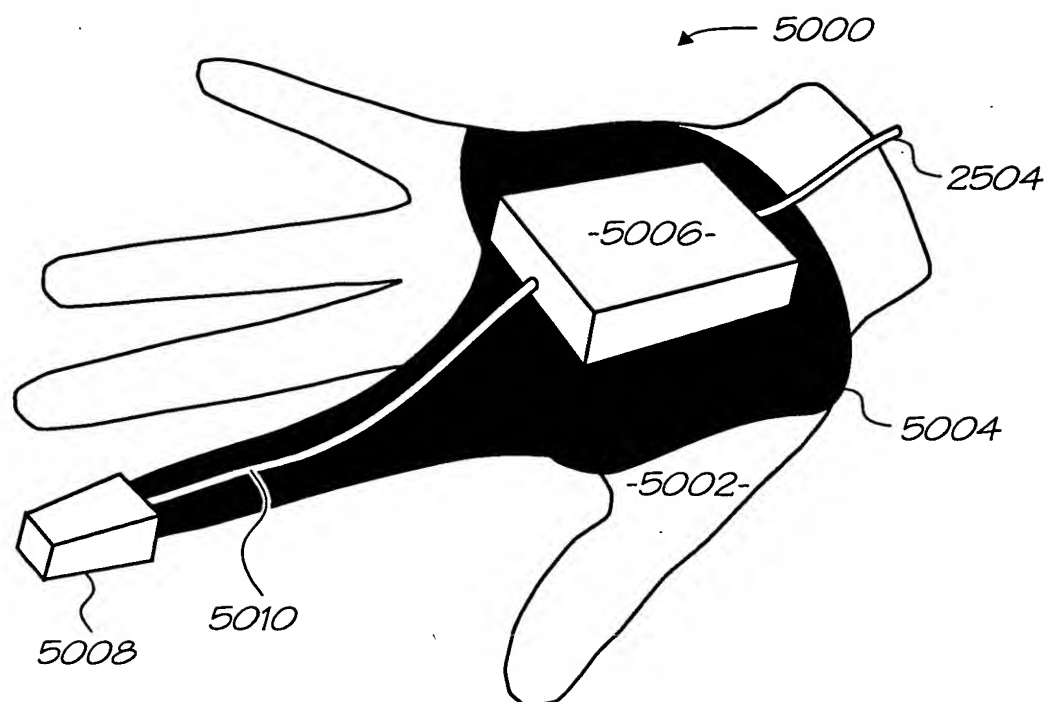


FIG. 70

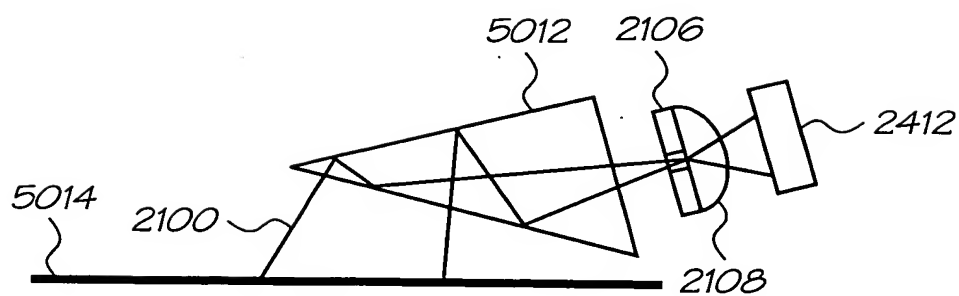


FIG. 71



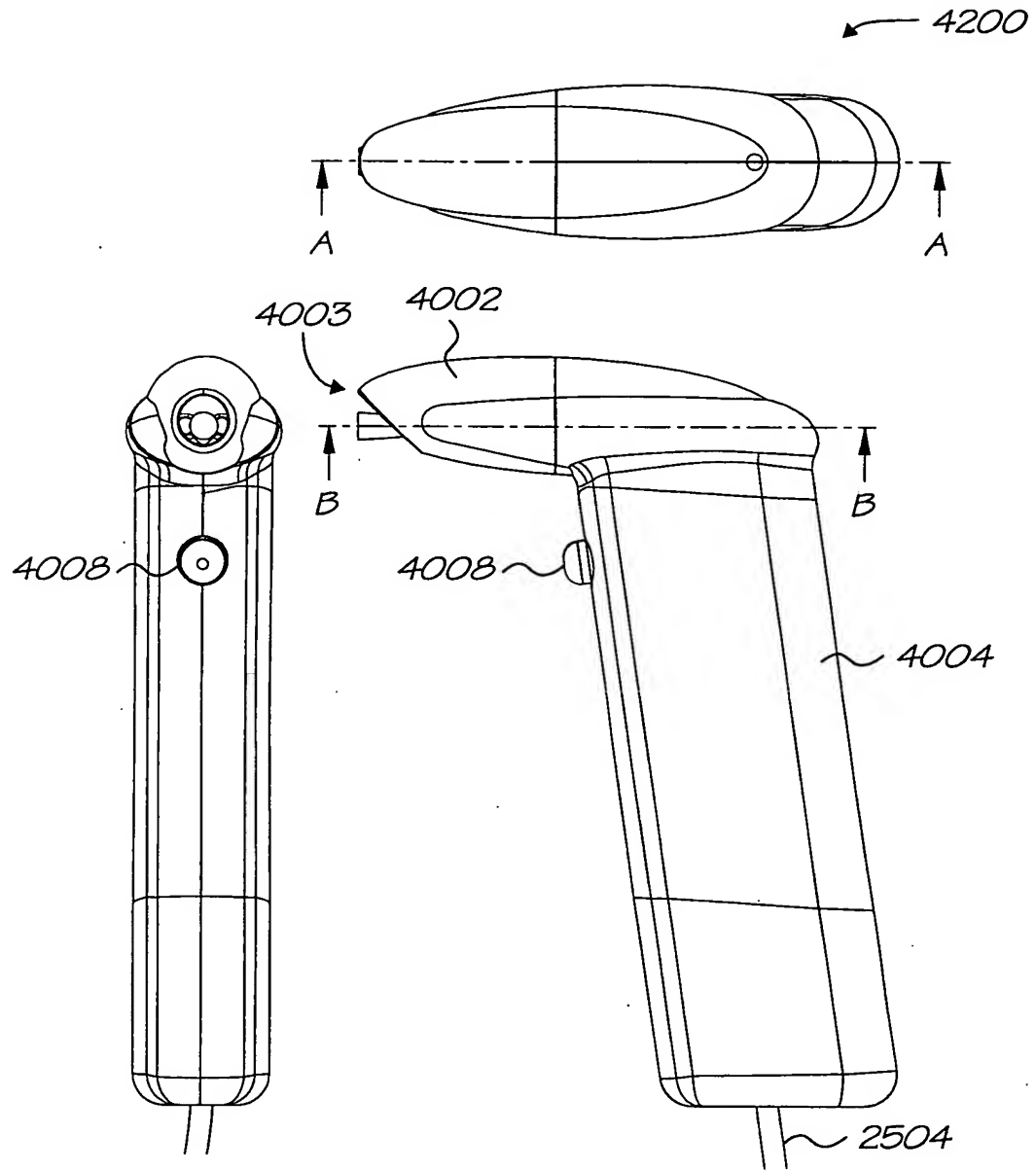


FIG. 72

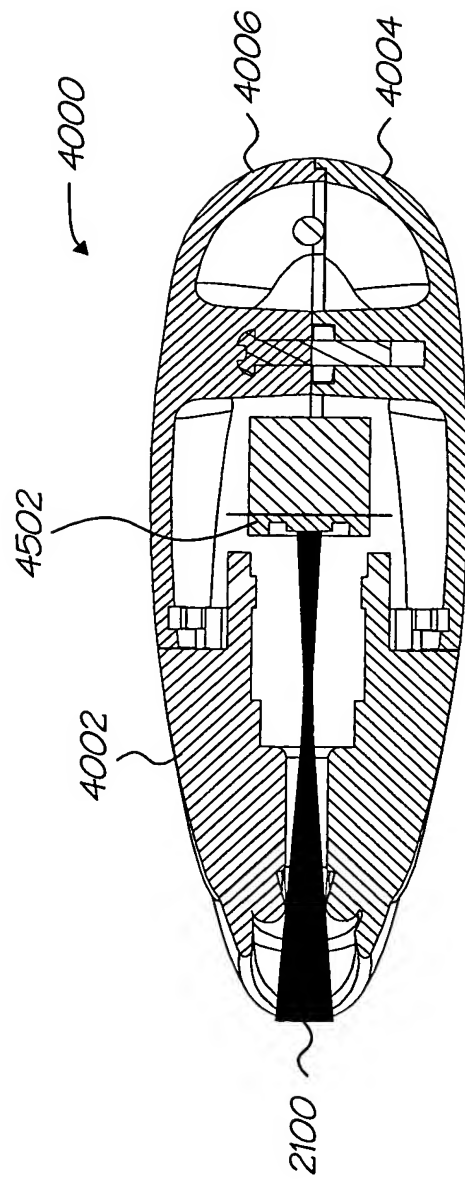


FIG. 73

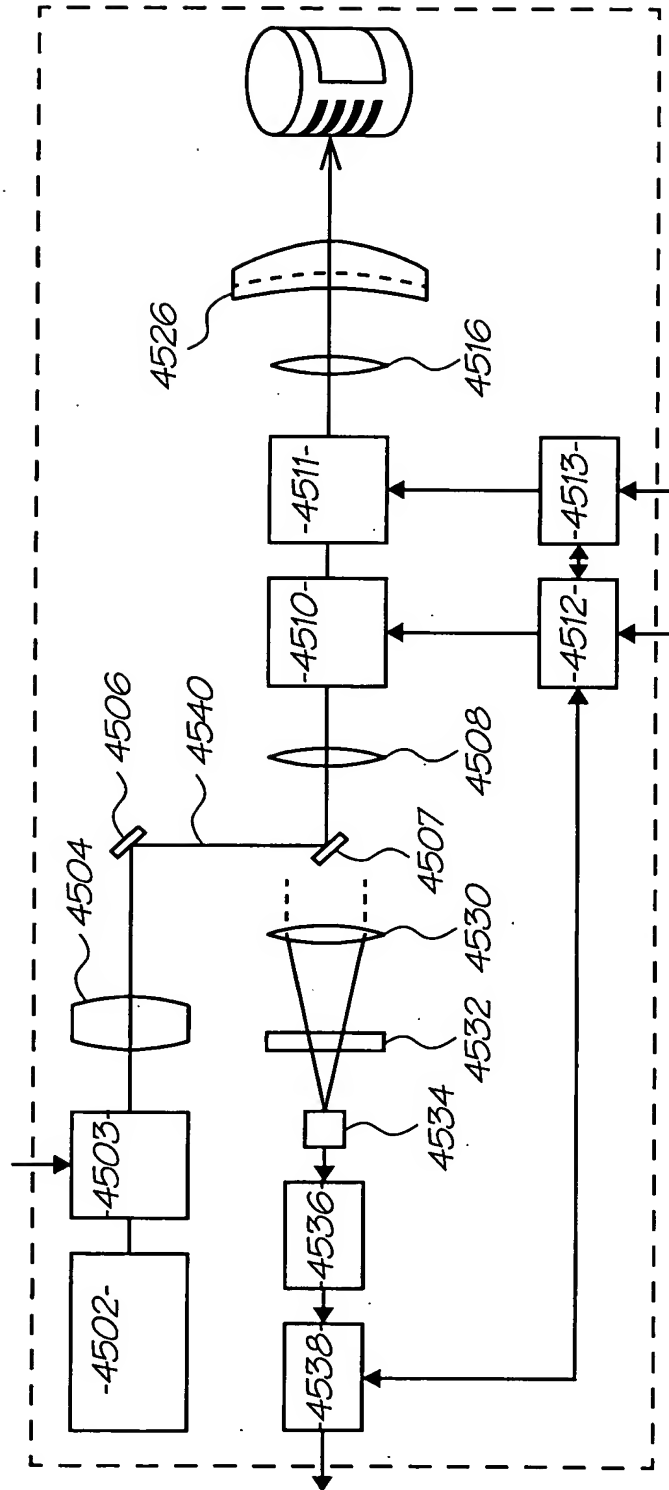


FIG. 74

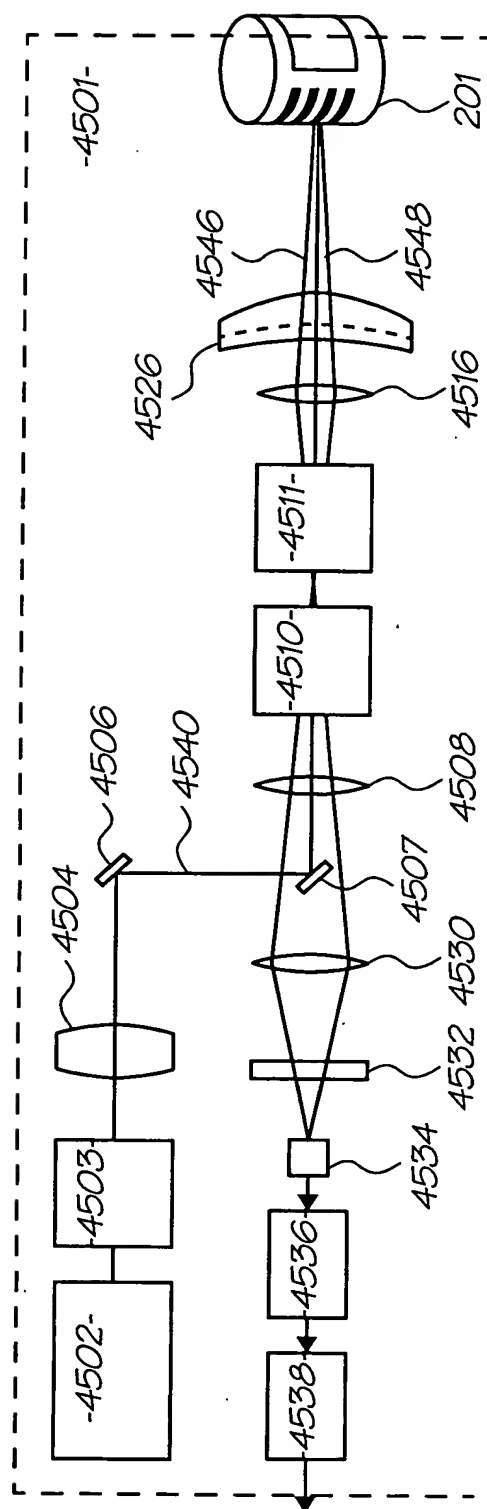


FIG. 75

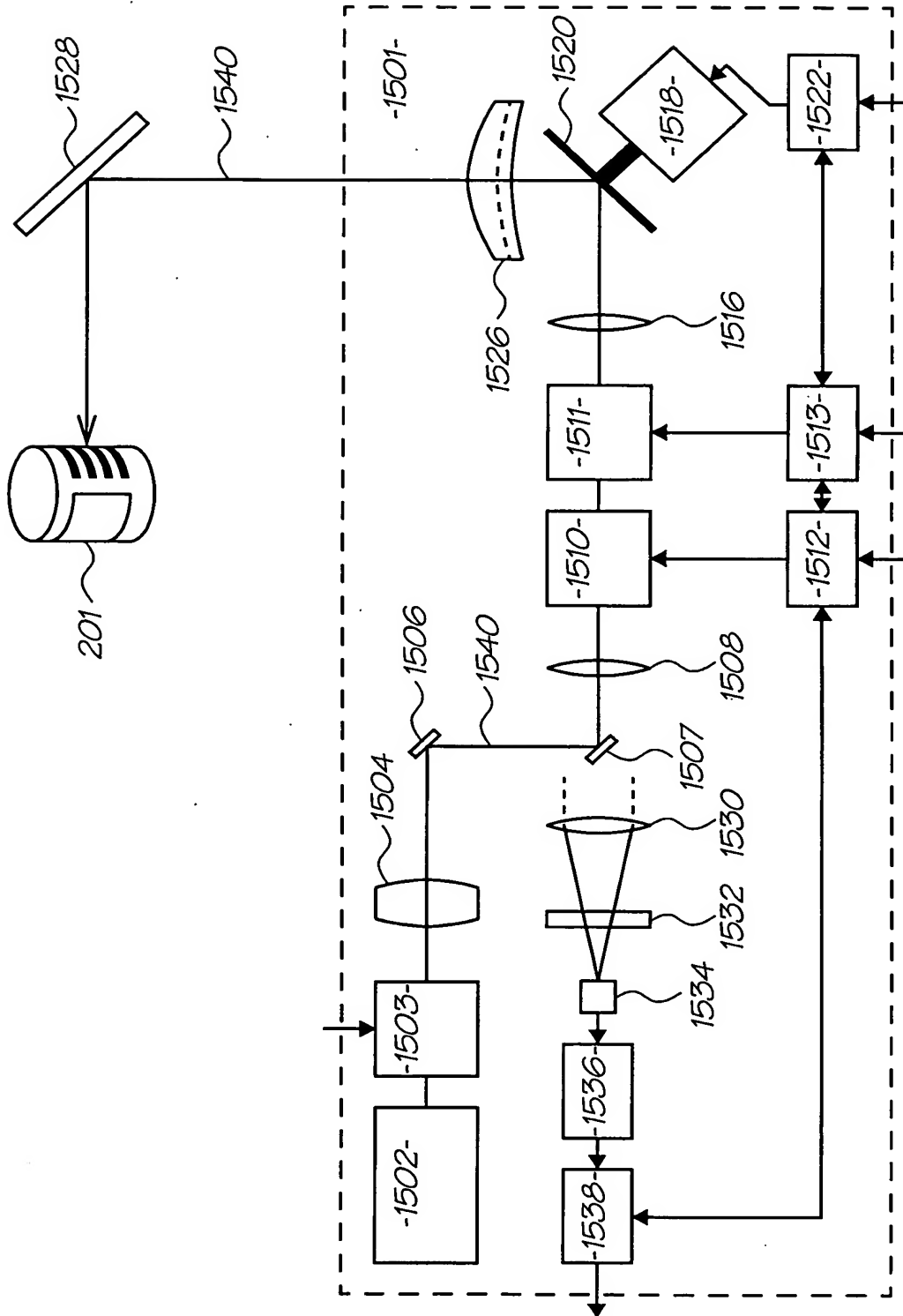


FIG. 76

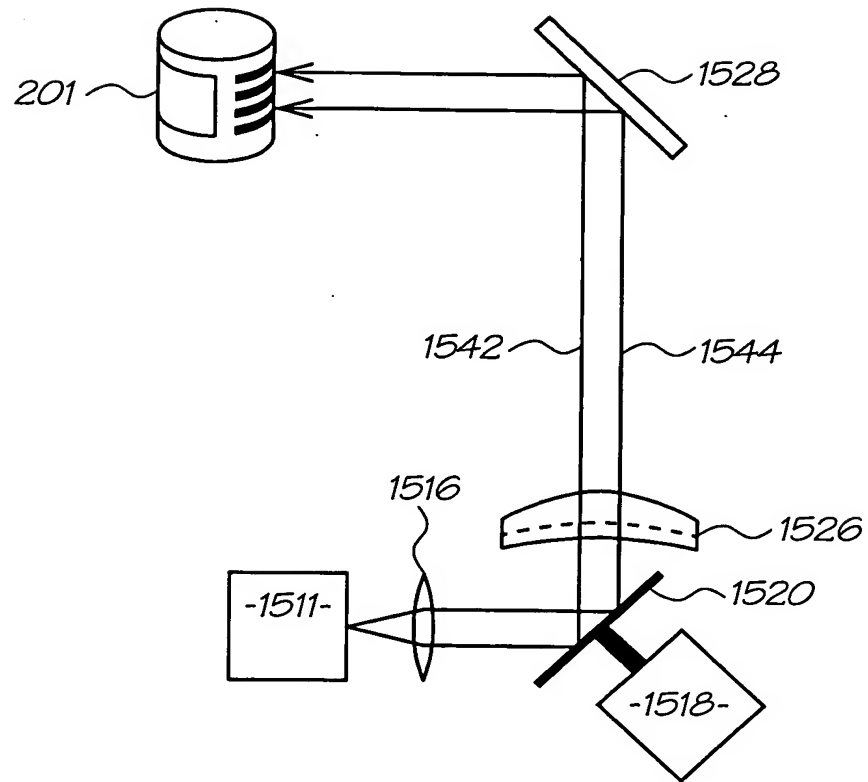


FIG. 77

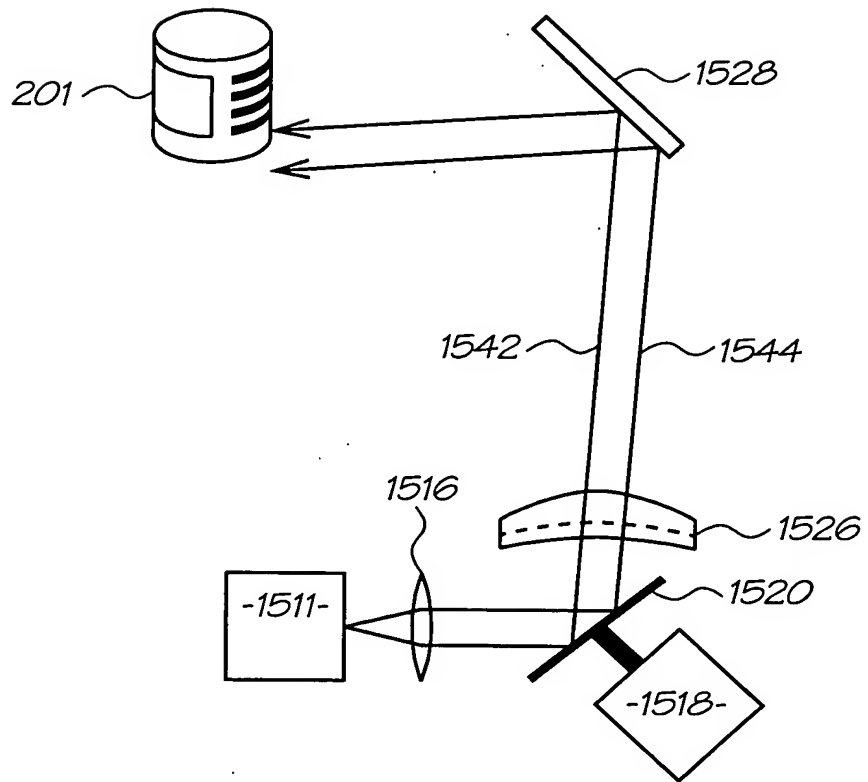


FIG. 78

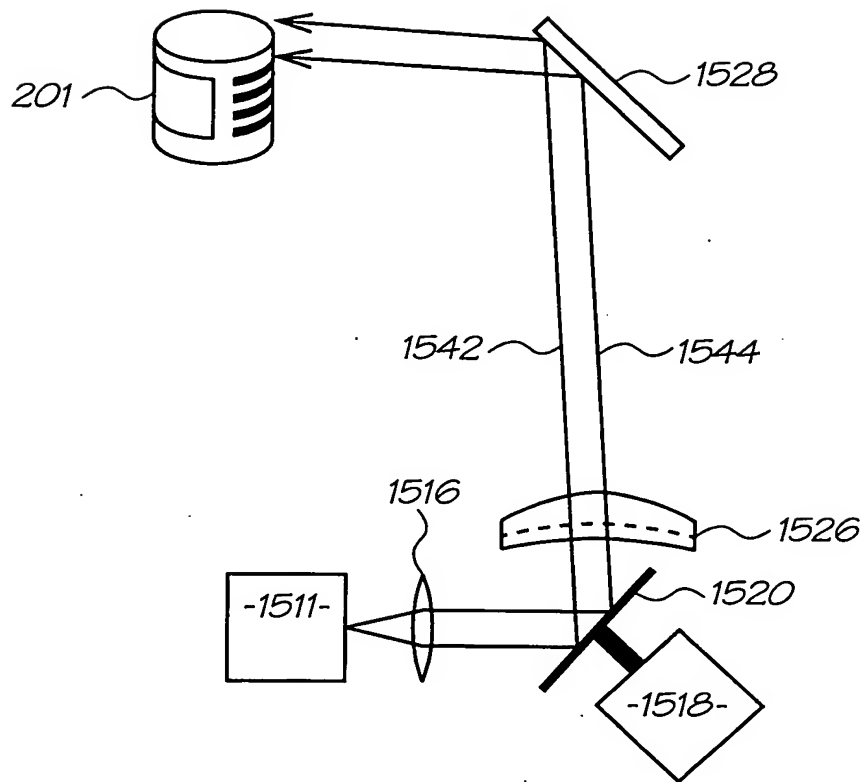


FIG. 79



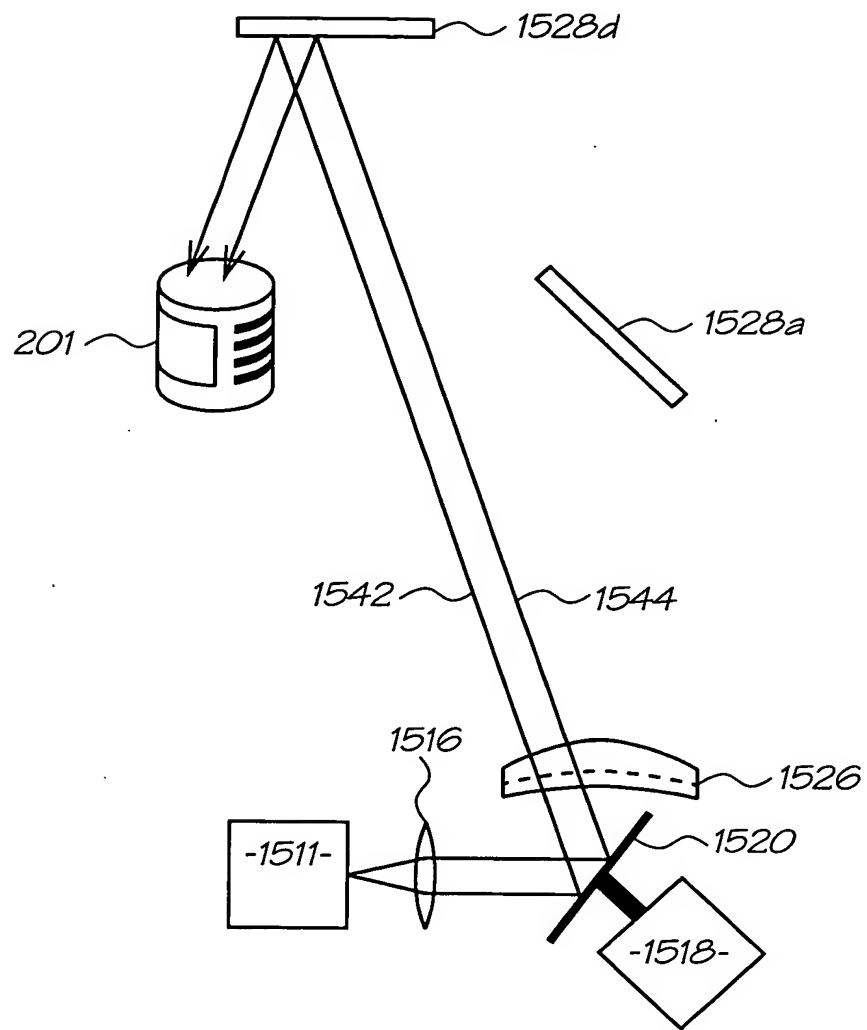


FIG. 80



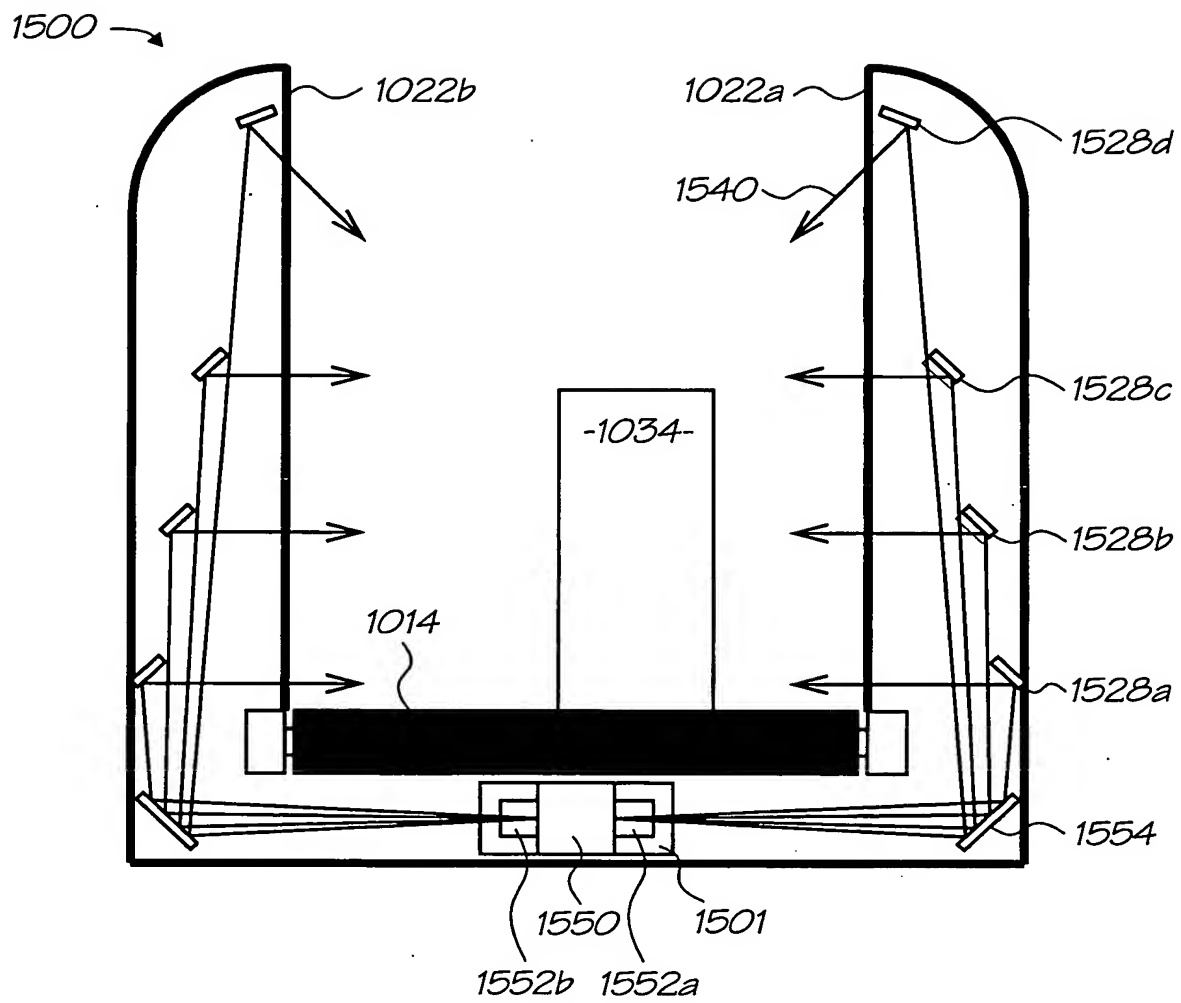


FIG. 82

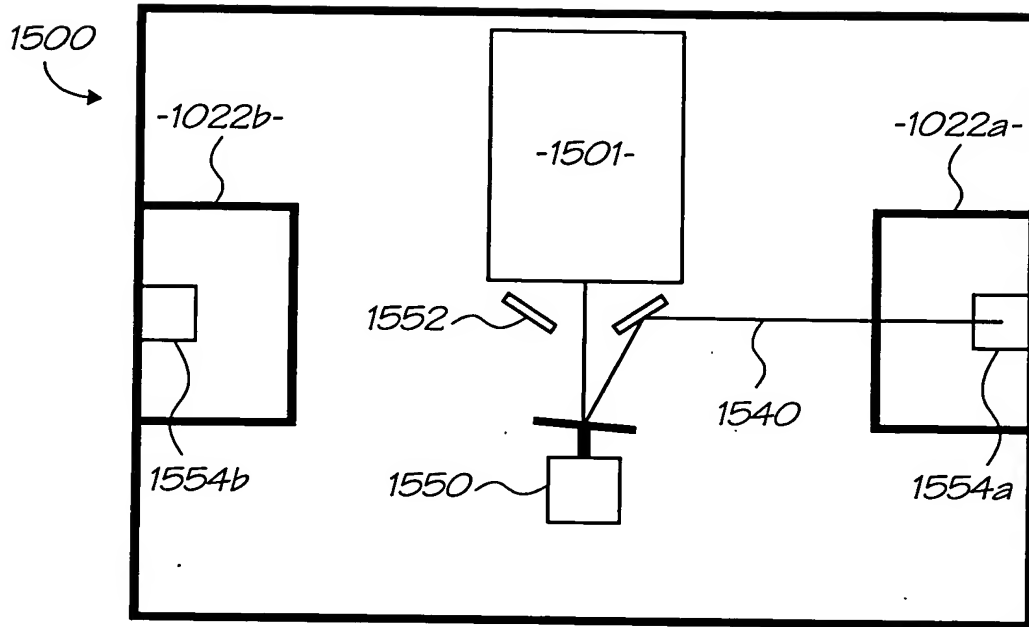


FIG. 83

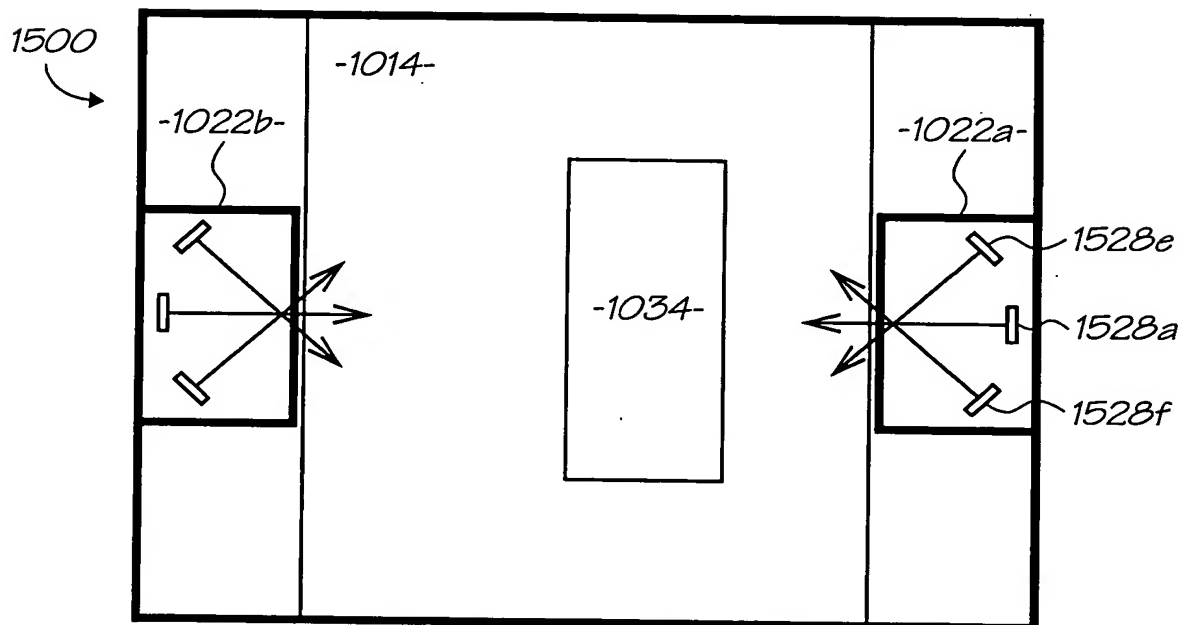


FIG. 84

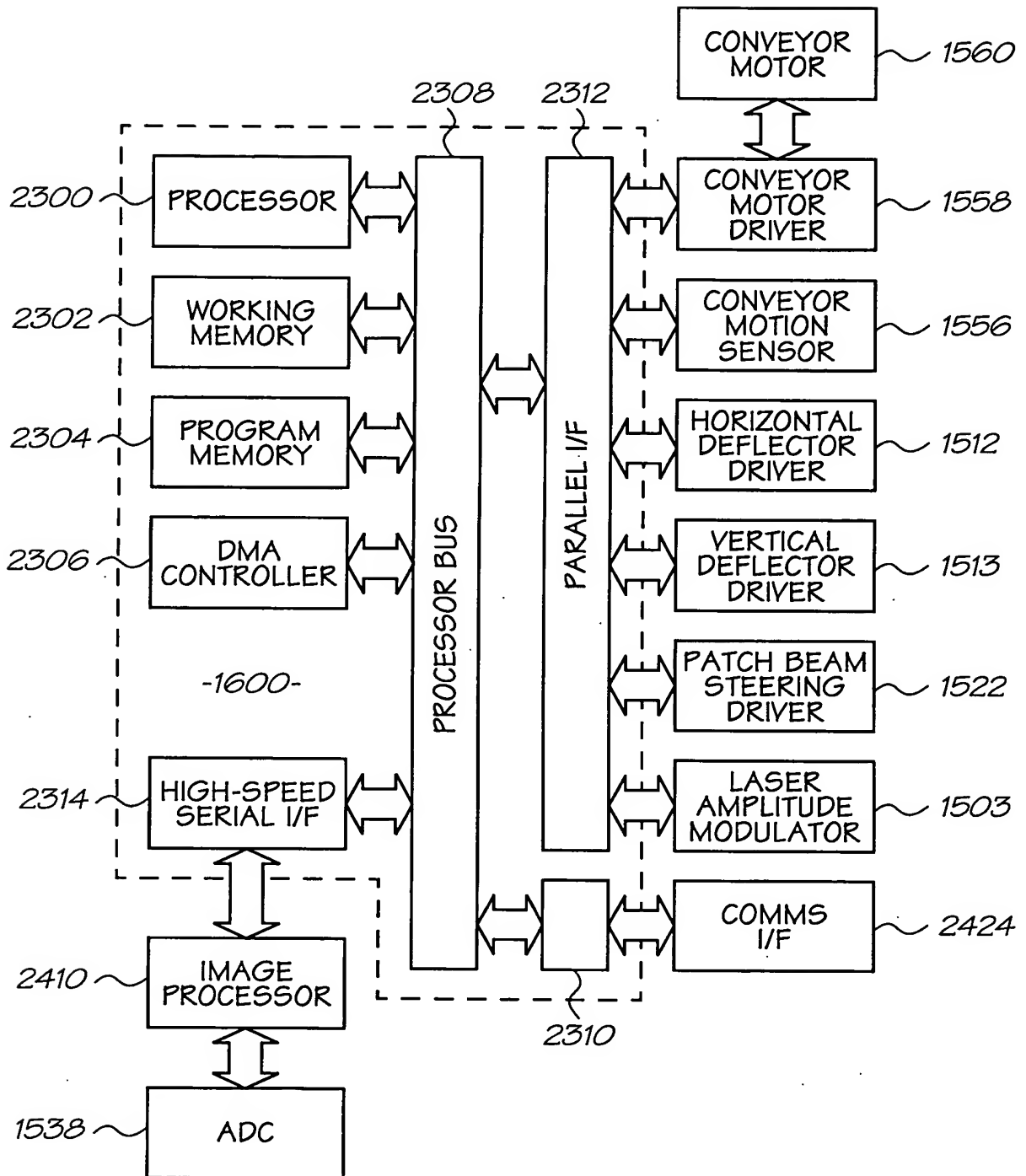


FIG. 85

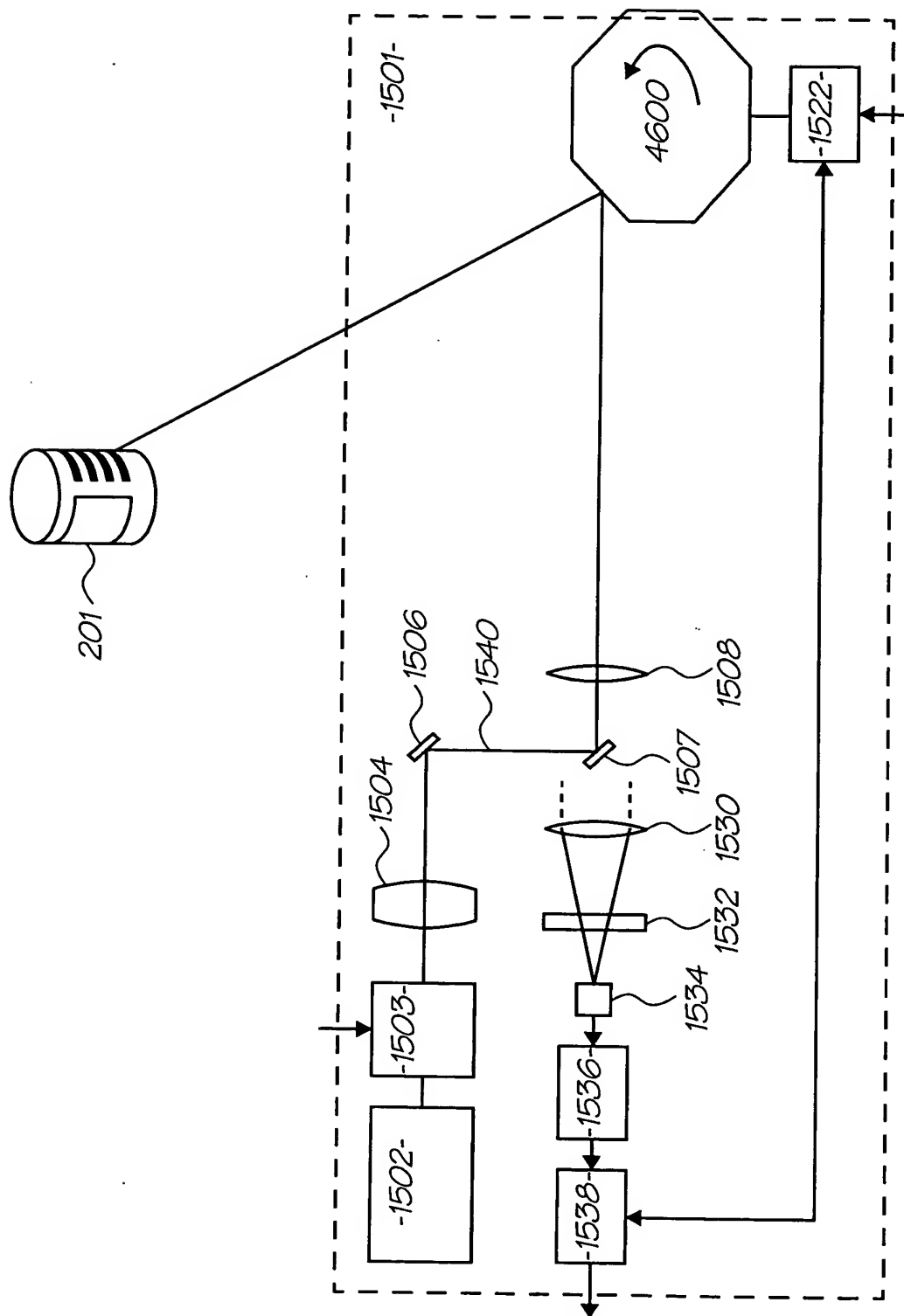


FIG. 86

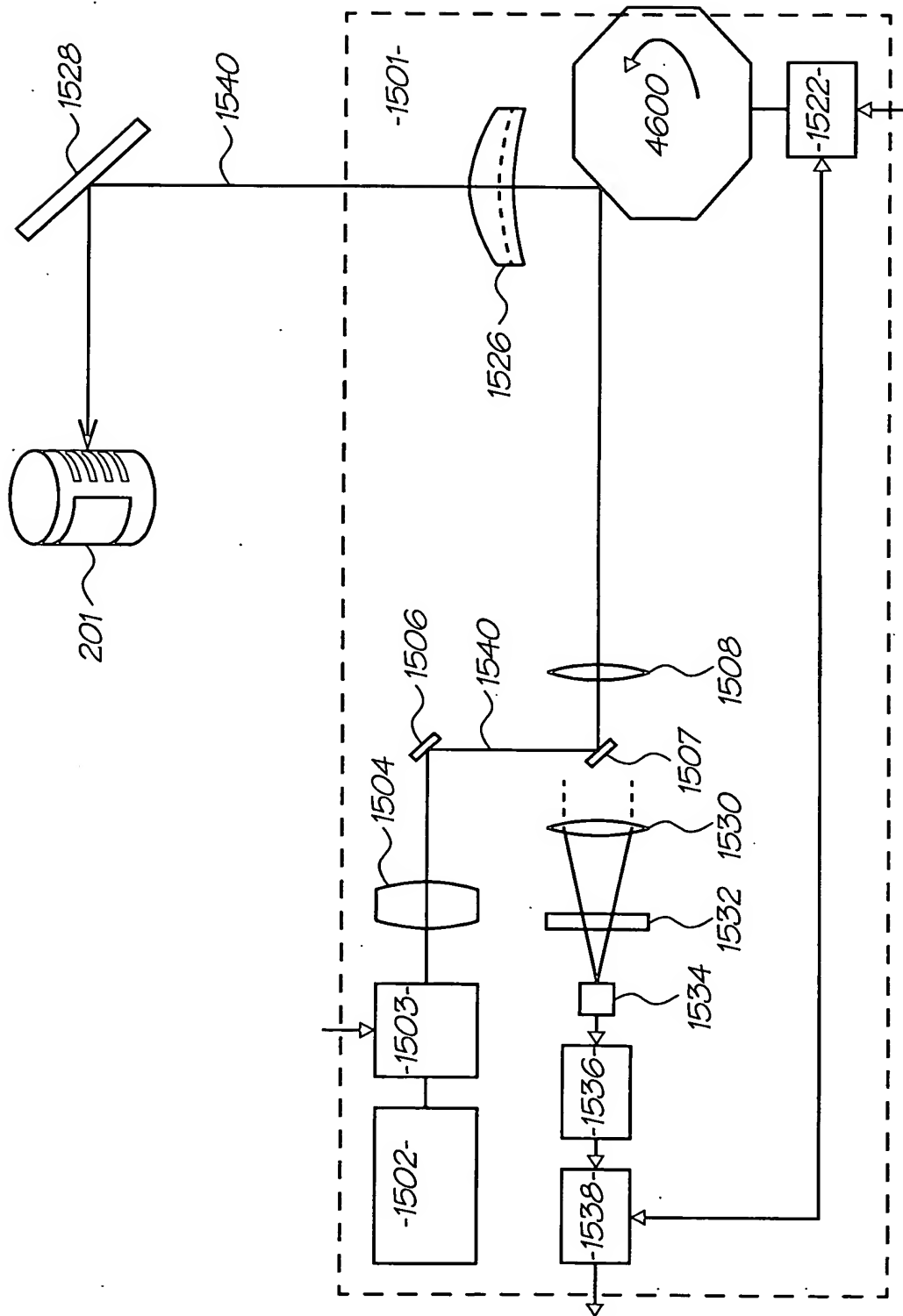


FIG. 87

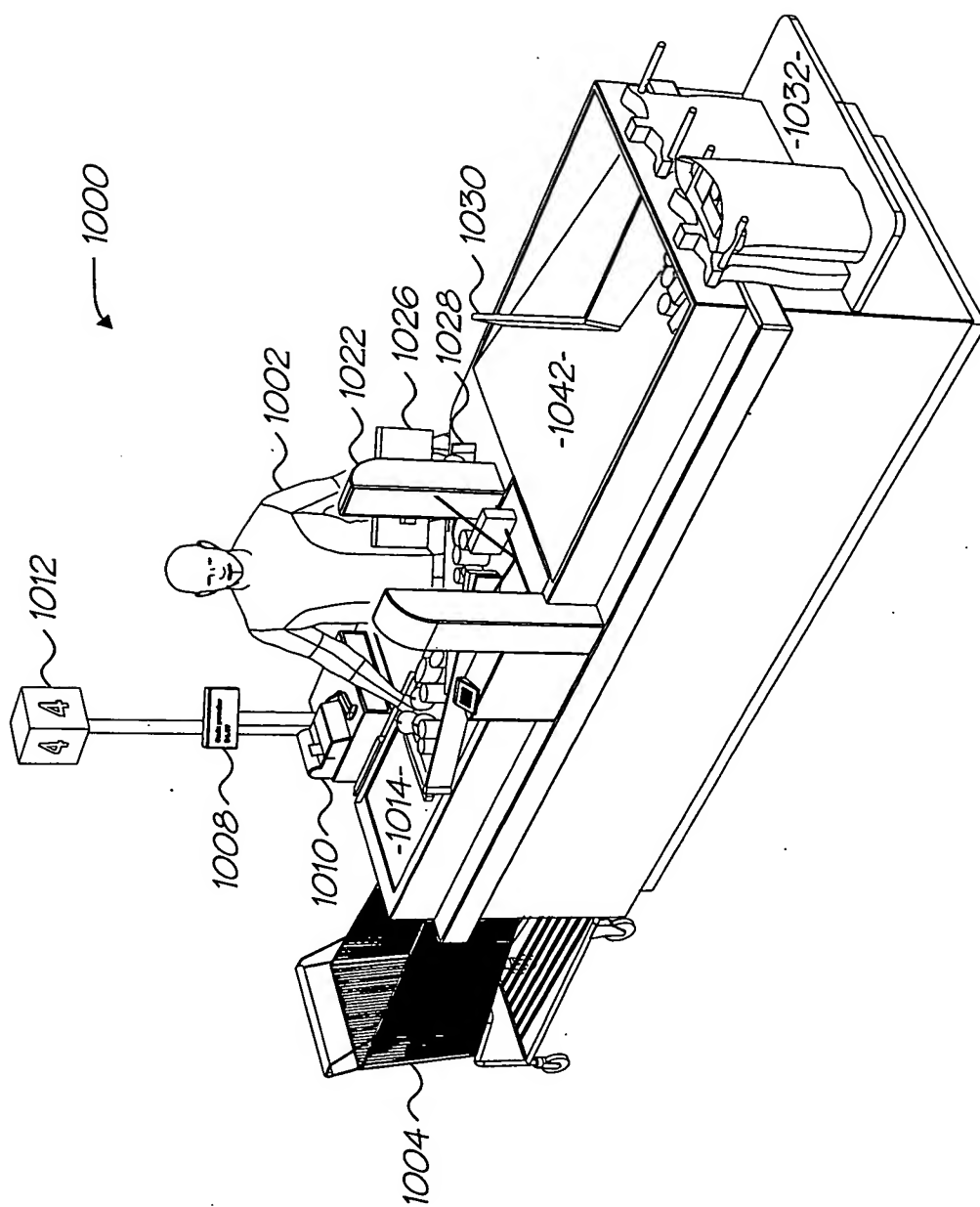


FIG. 88



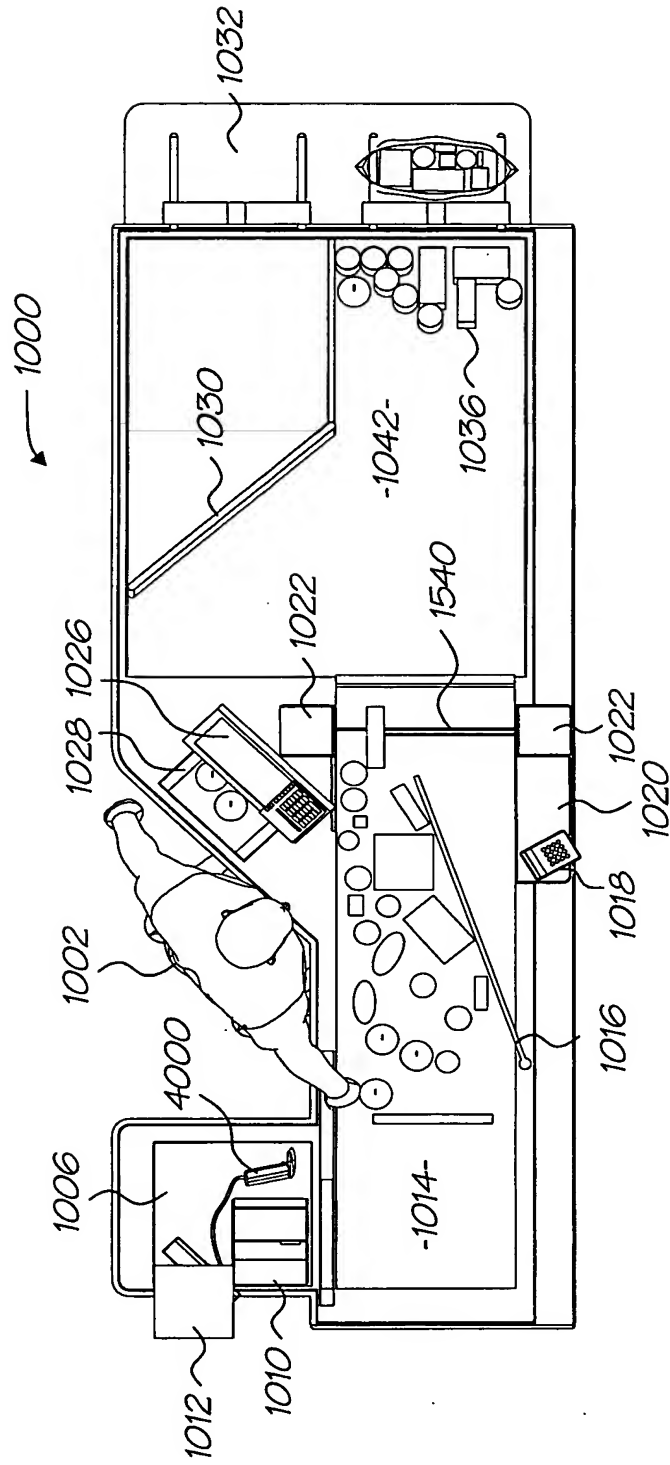


FIG. 89

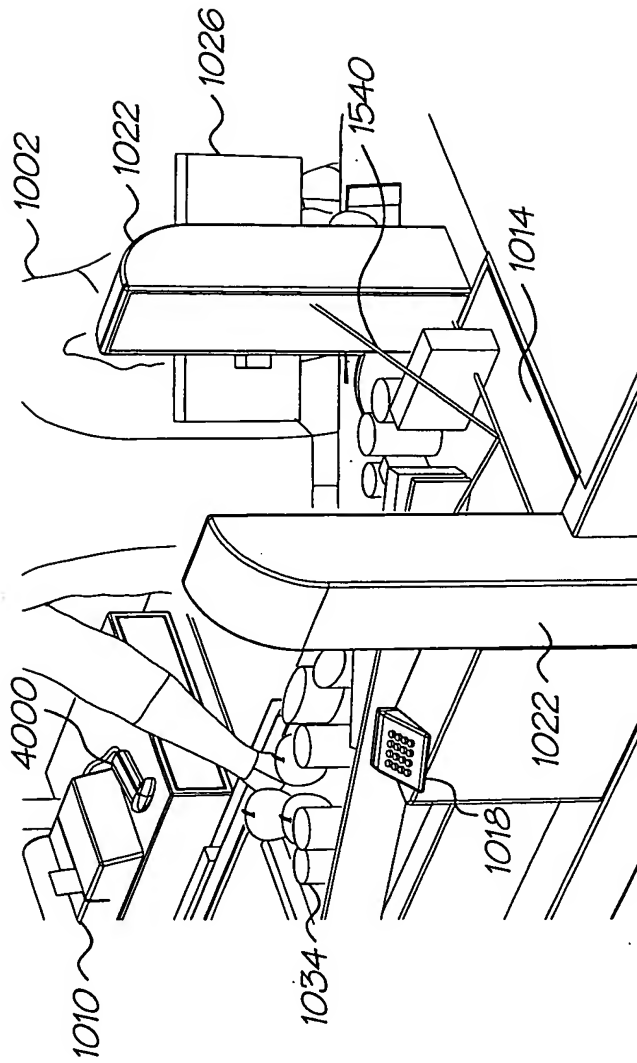


FIG. 90

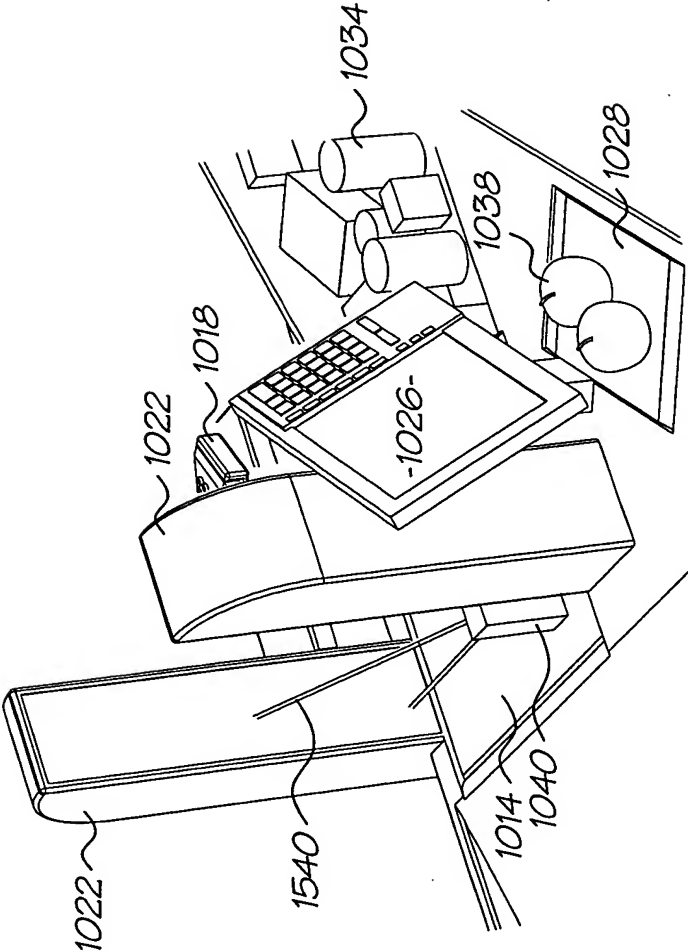


FIG. 91

80/157

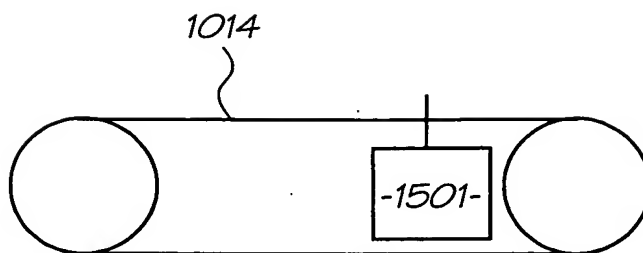


FIG. 92

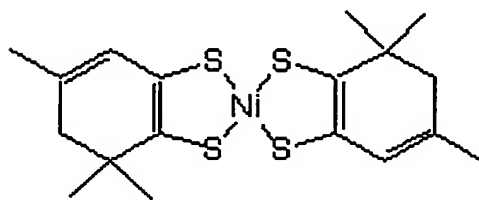


FIG. 93

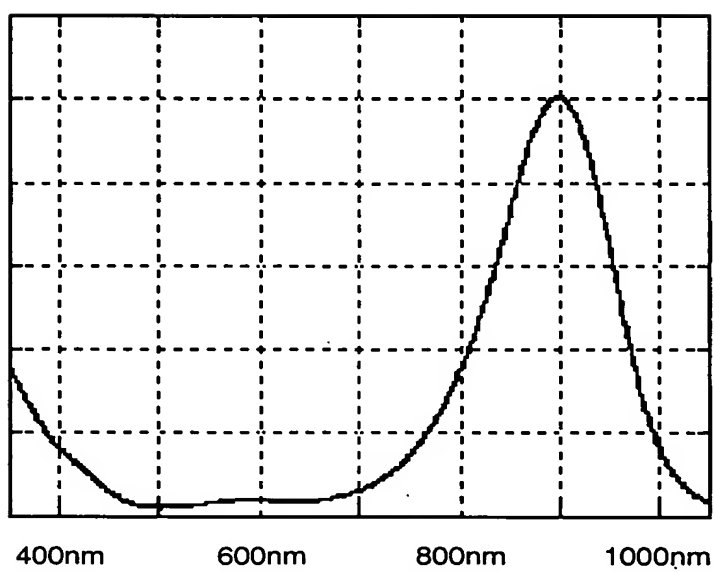
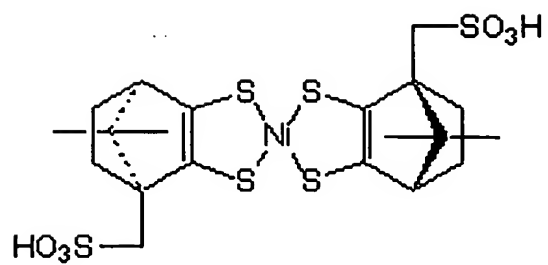
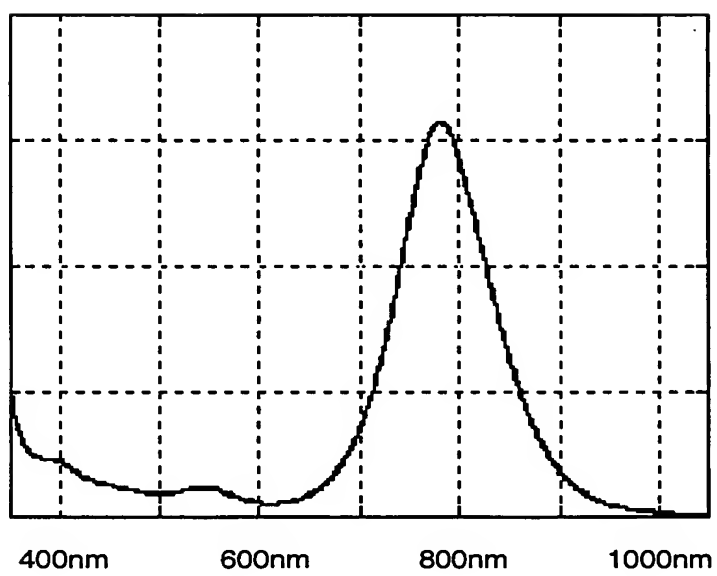


FIG. 94

81/157



*FIG. 95*



*FIG. 96*

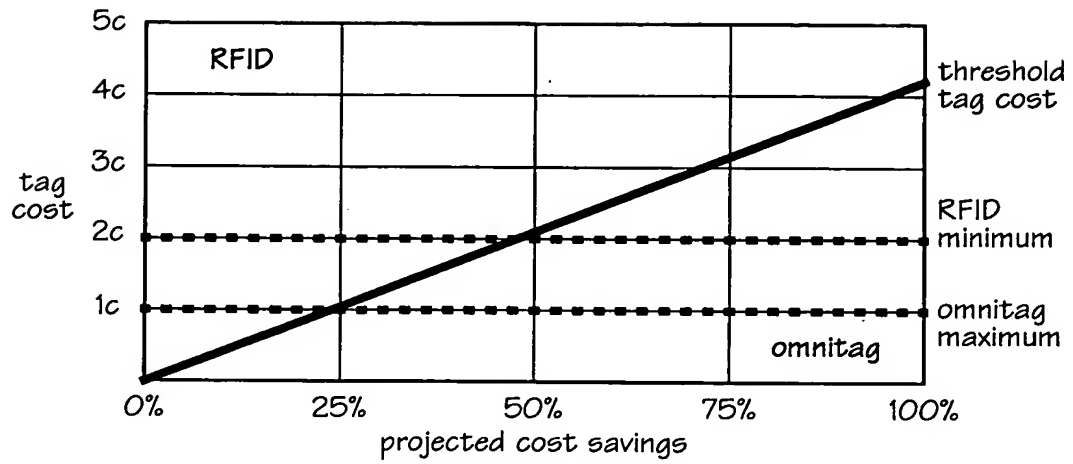


FIG. 97

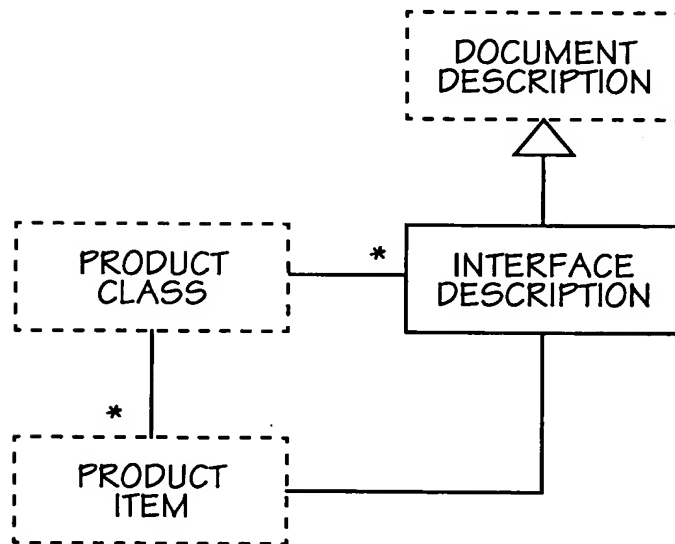


FIG. 98

83/157

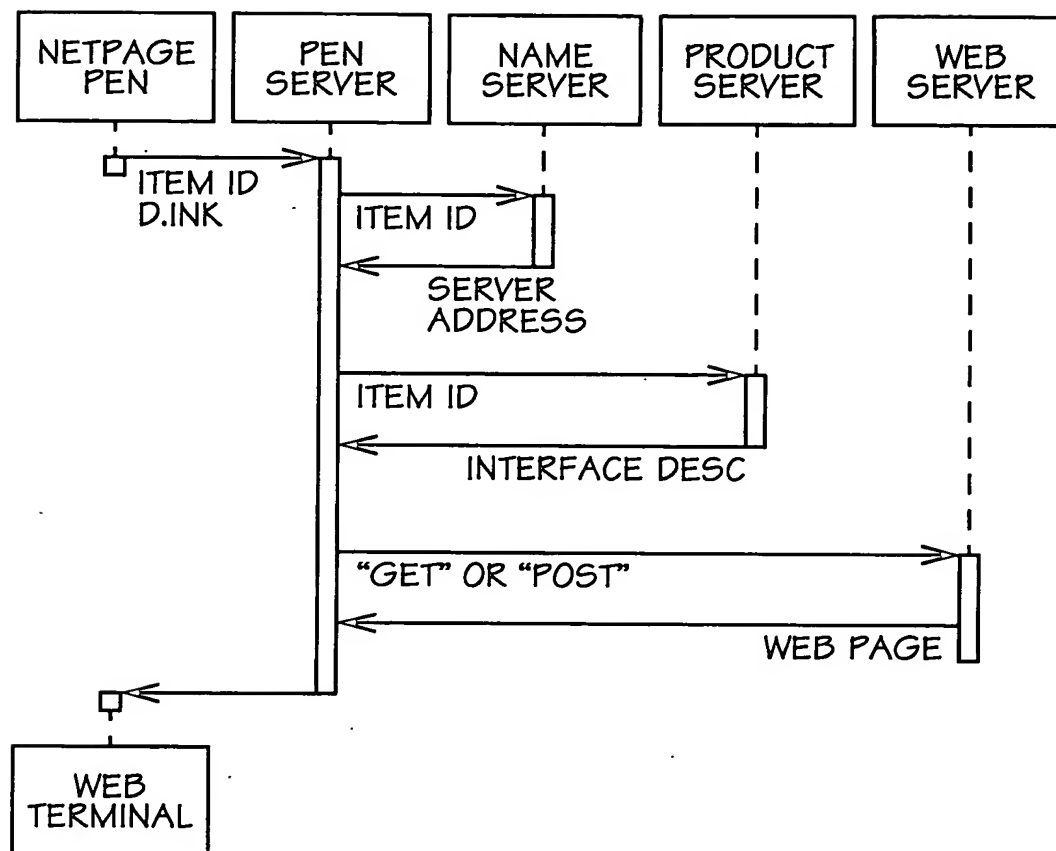


FIG. 99

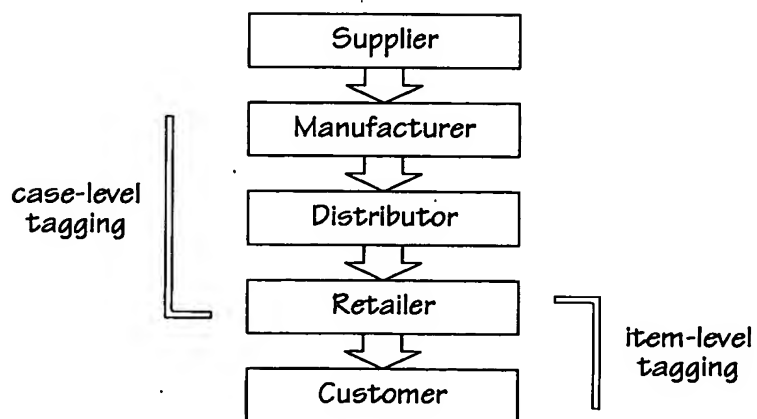


FIG. 100

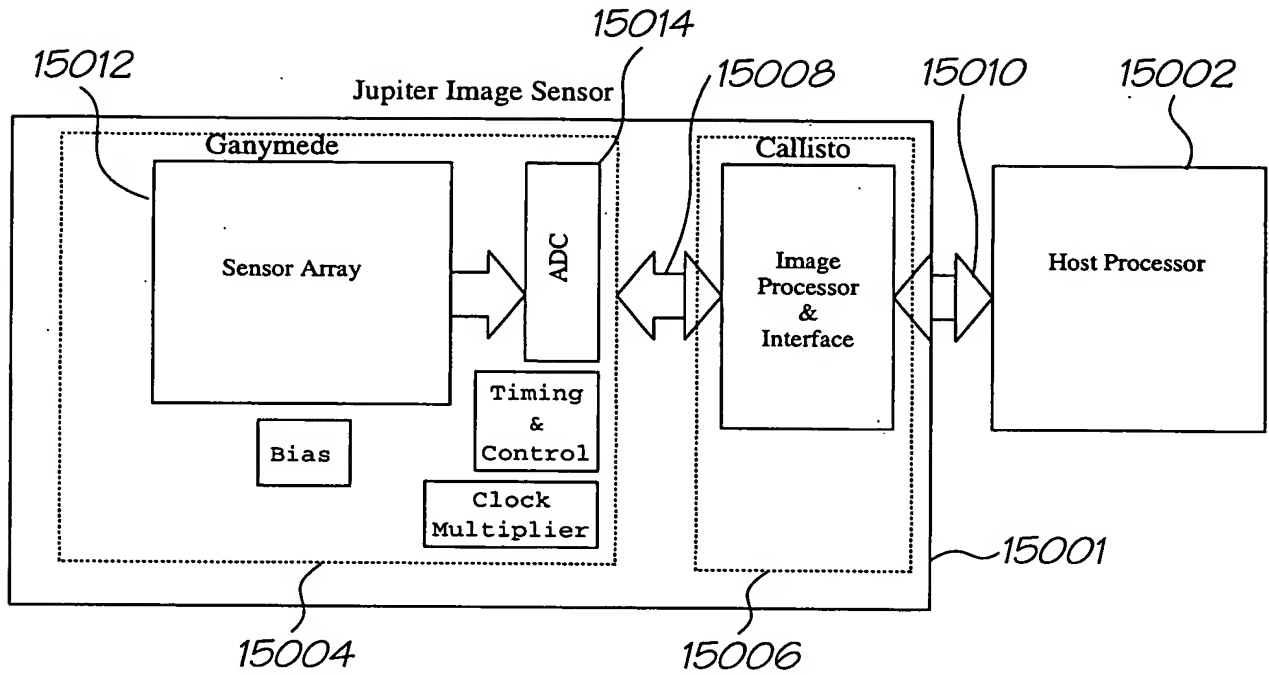


FIG. 101

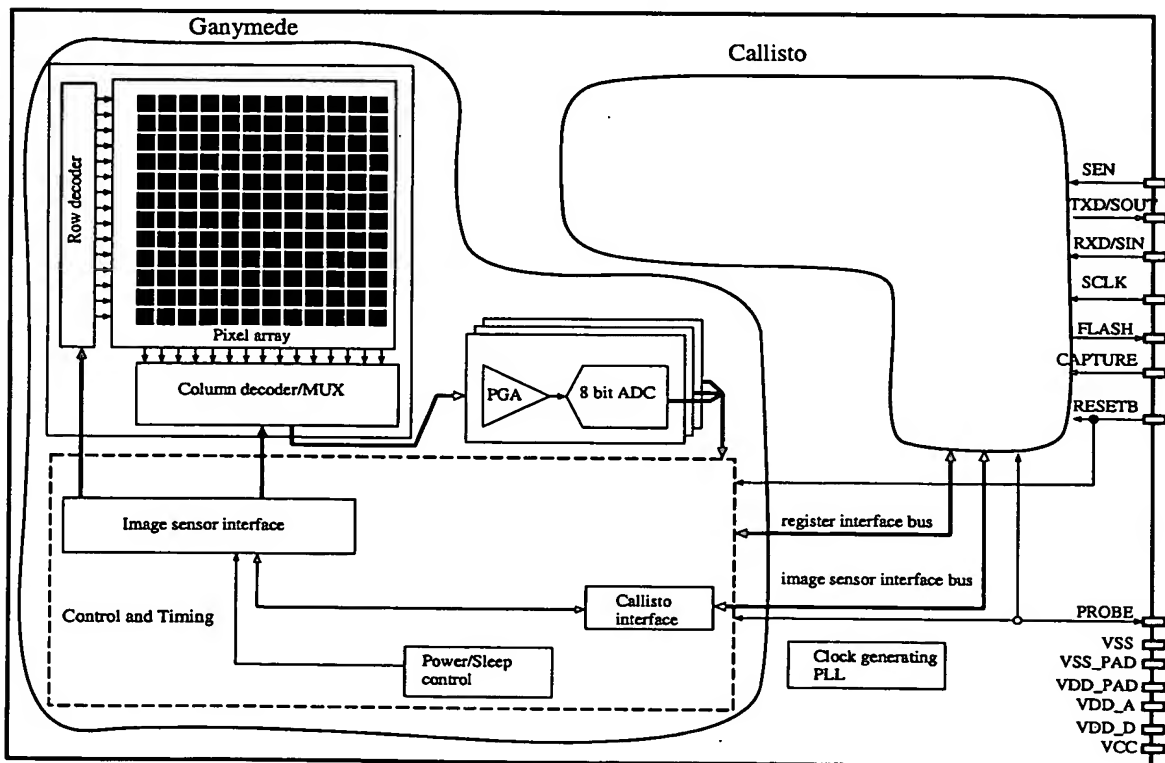


FIG. 102



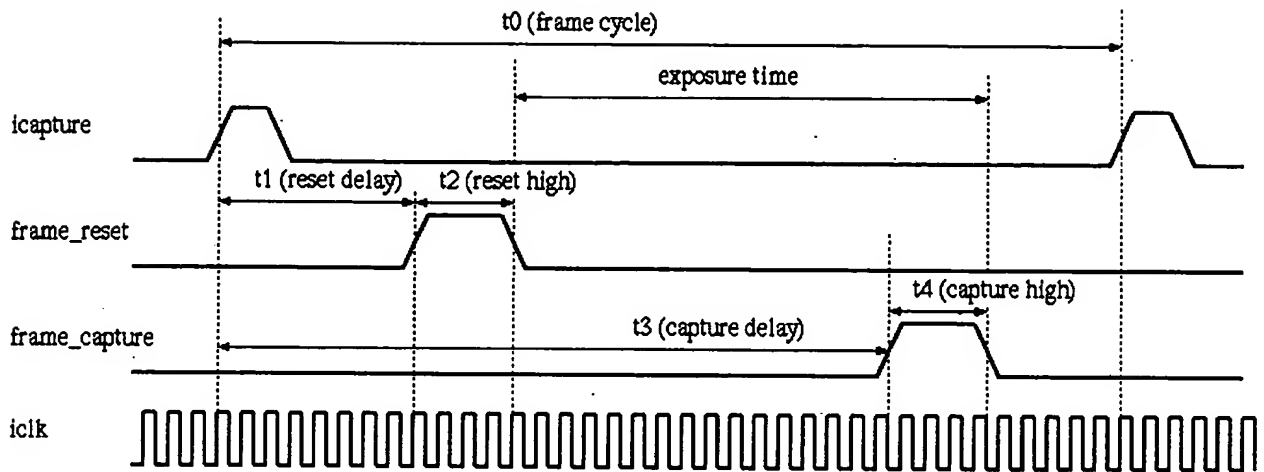
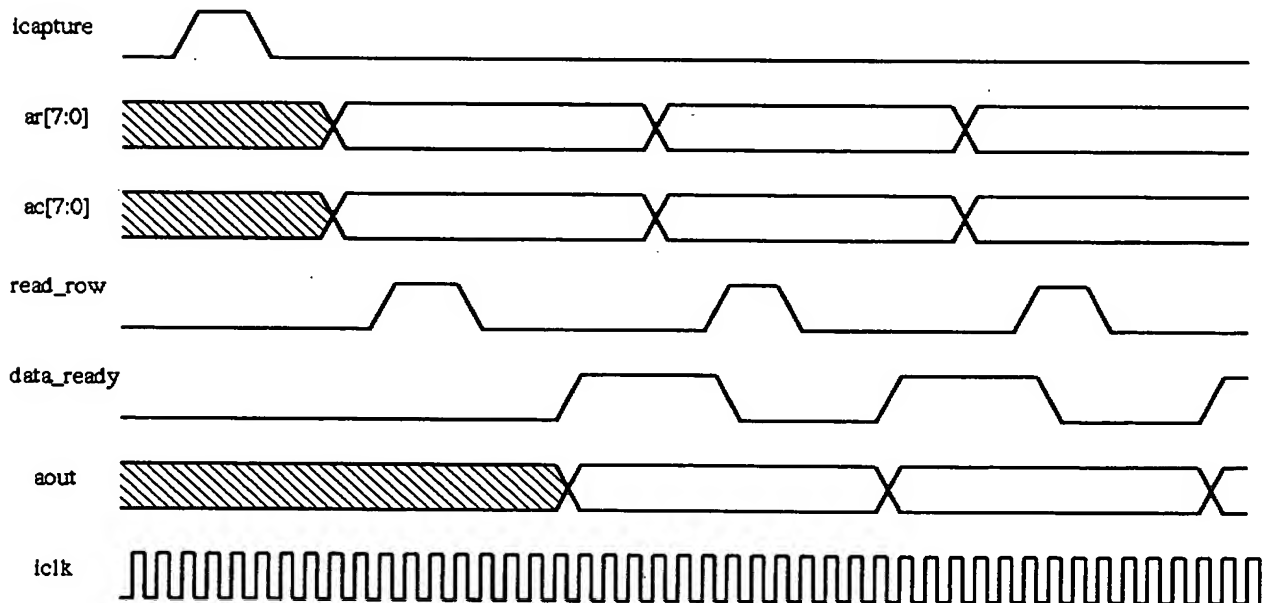


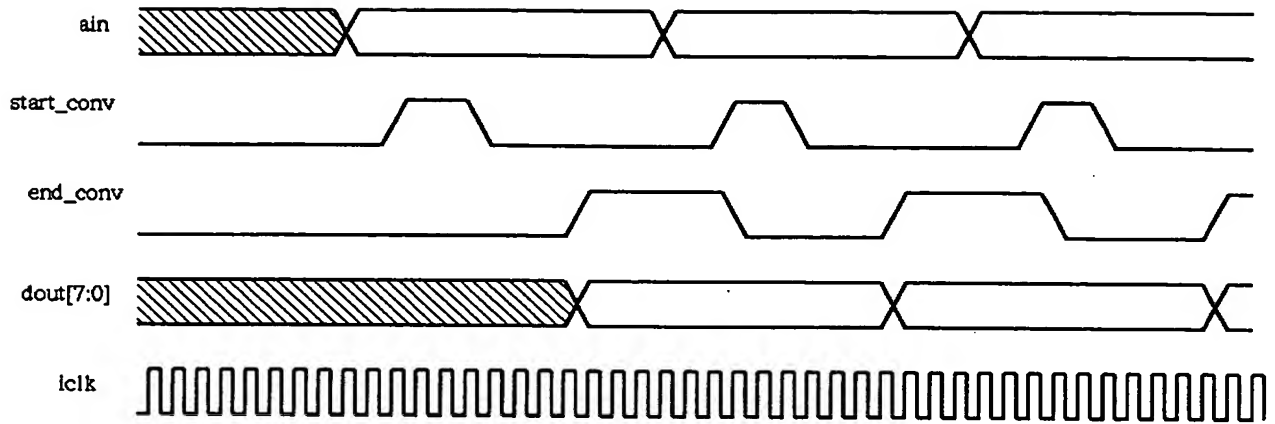
FIG. 103



Note: the number of clock pulses between events is indicative only. The actual values will depend on the particular implementation.

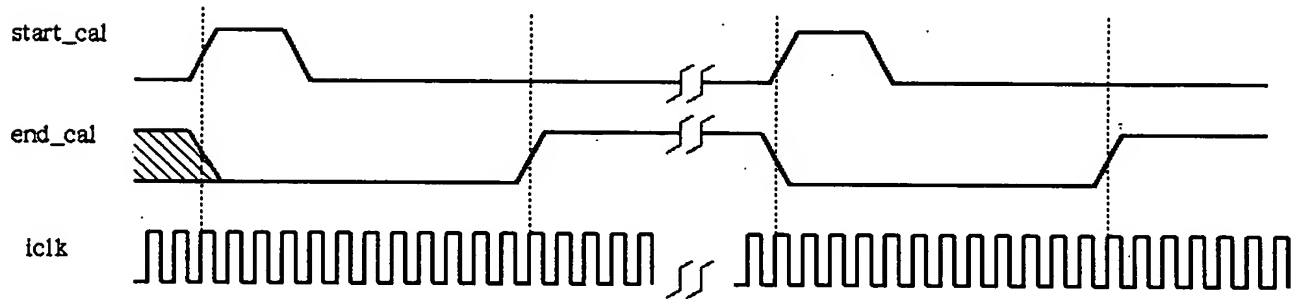
FIG. 104

86/157



Note: the number of clock pulses between events is indicative only.

FIG. 105



Note: the number of clock pulses between events is indicative only. The actual values will depend on the particular implementation.

FIG. 106

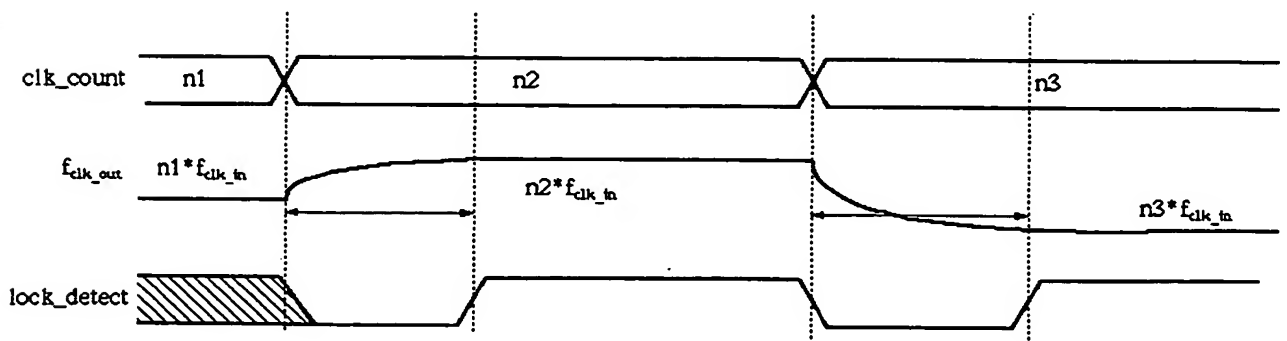


FIG. 107

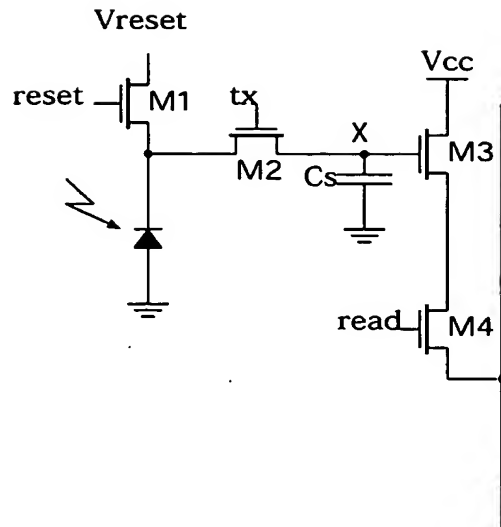


FIG. 108a  
(PRIOR ART)

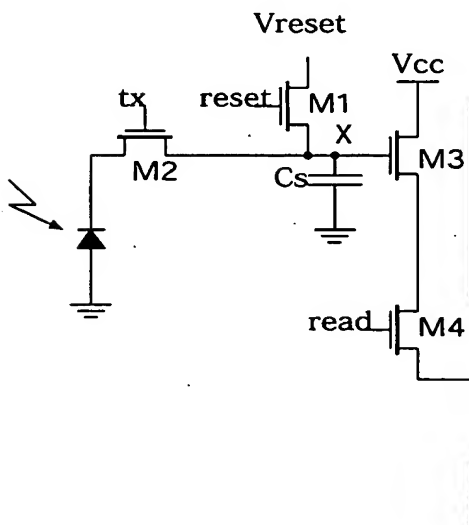


FIG. 108b  
(PRIOR ART)

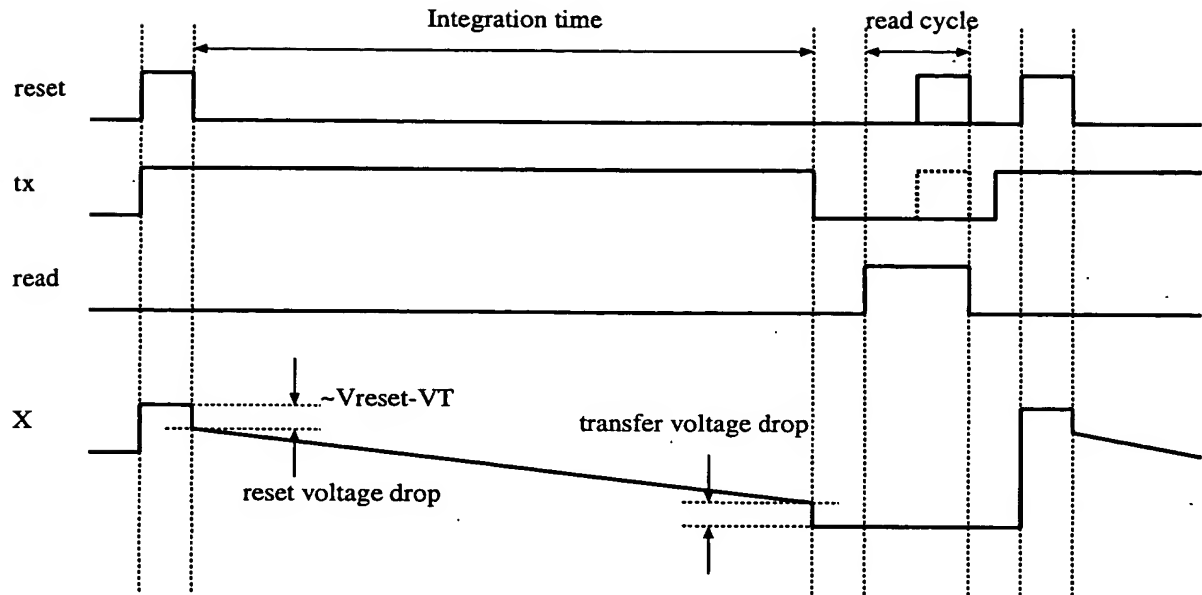


FIG. 109  
(PRIOR ART)

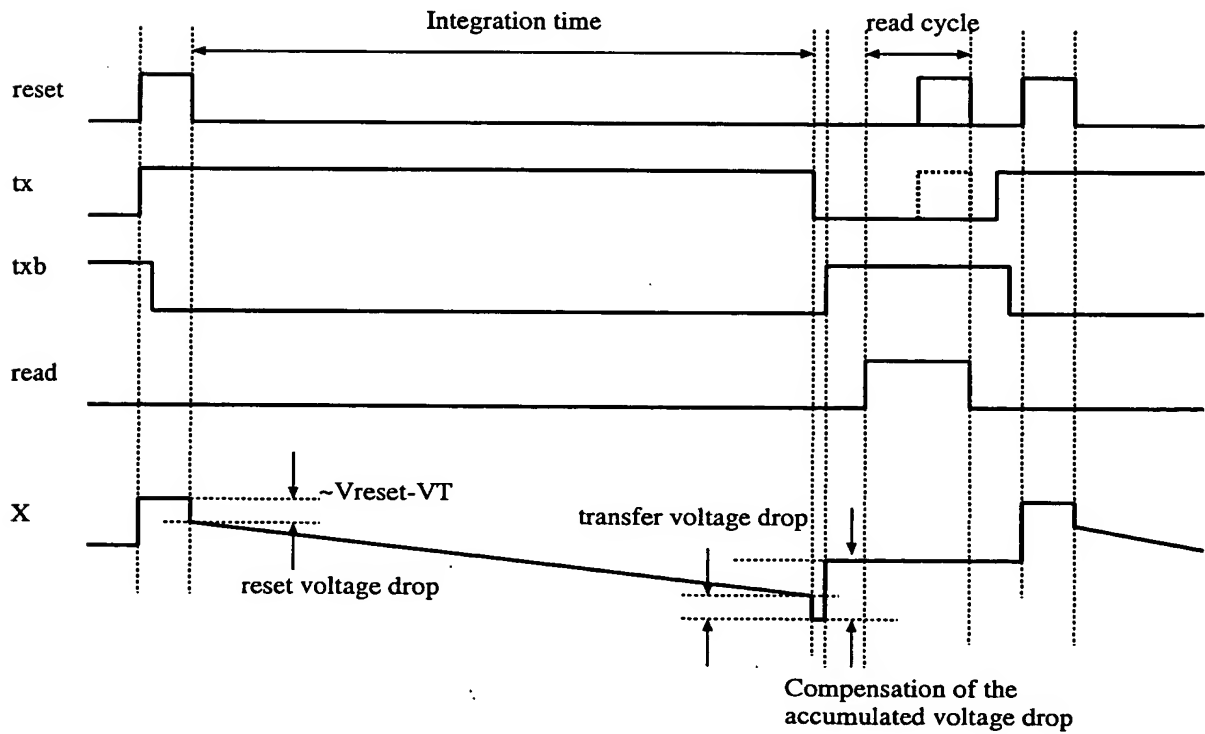


FIG. 109a

89/157

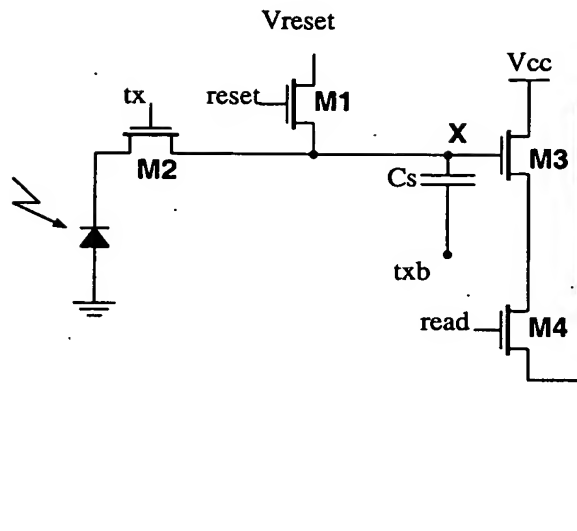


FIG. 110

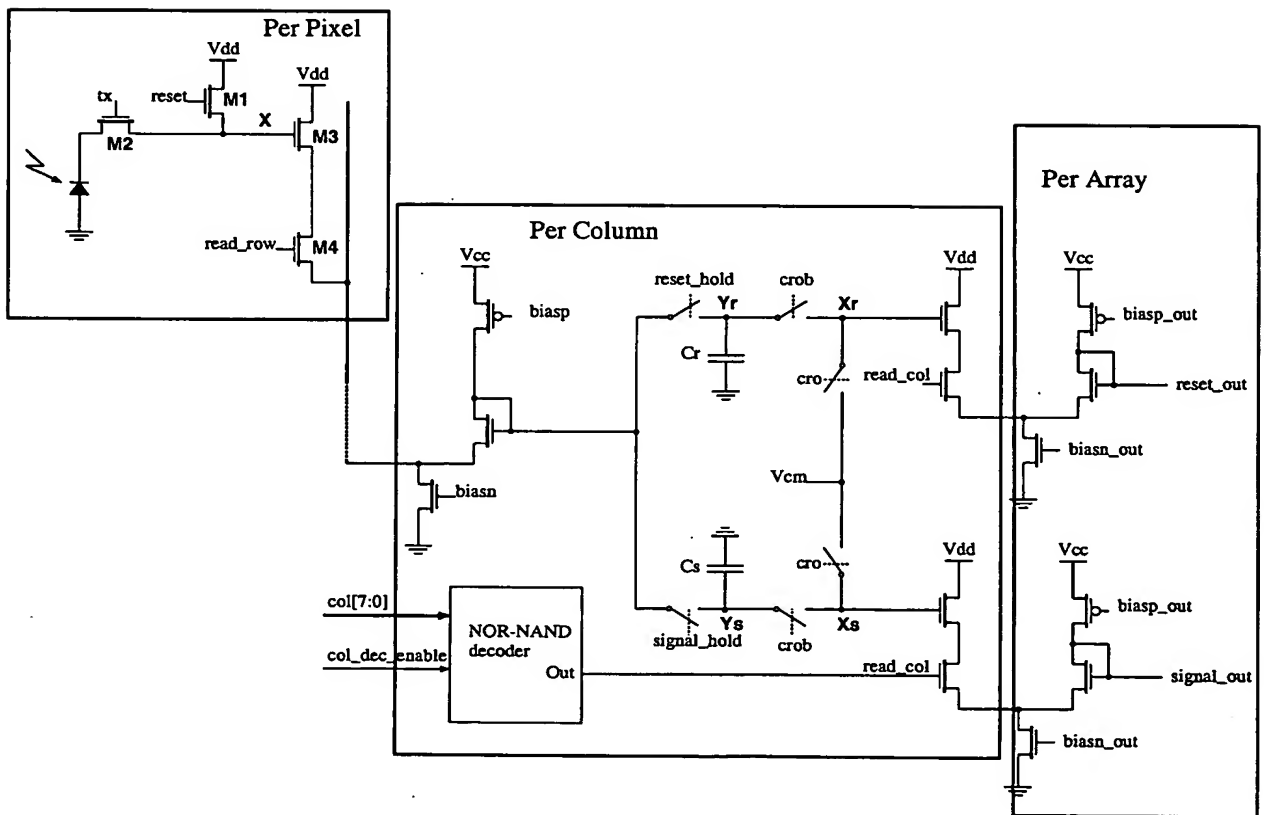


FIG. 111

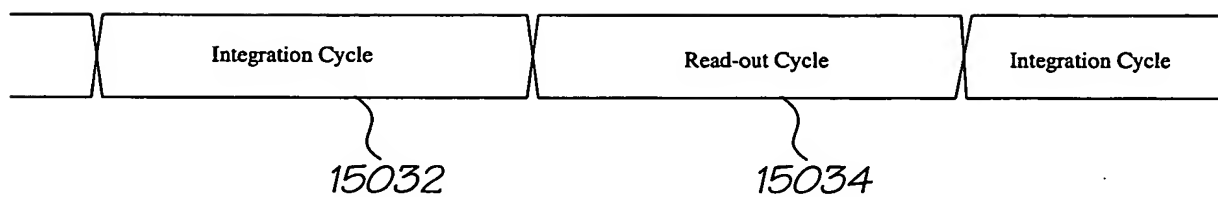


FIG. 111a

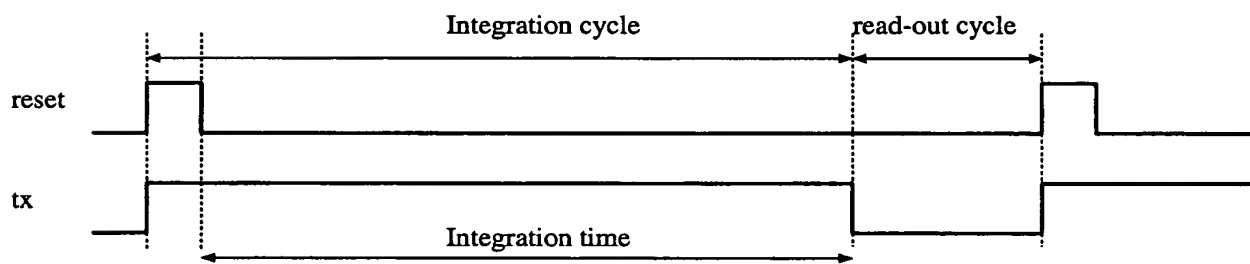


FIG. 112

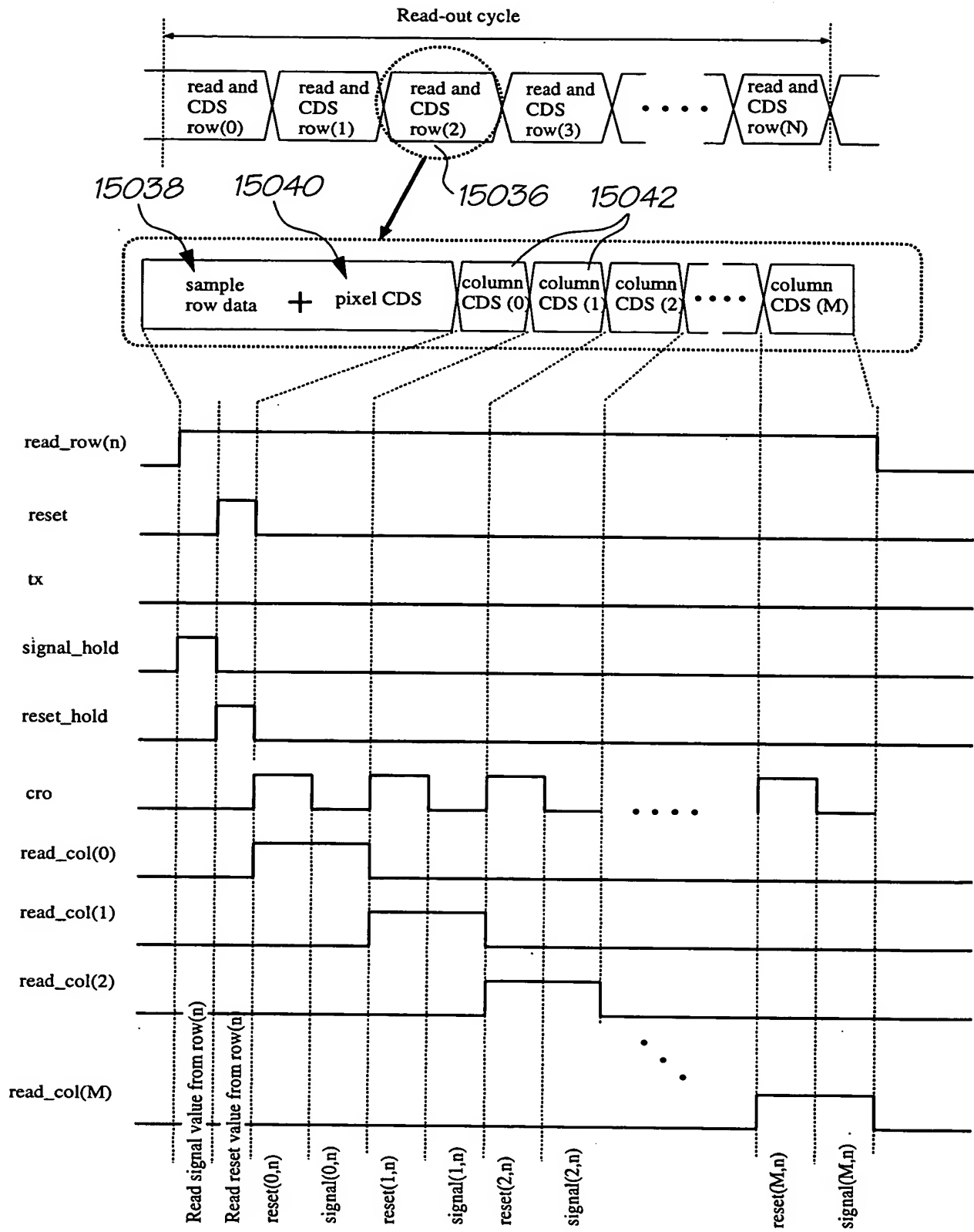


FIG. 113

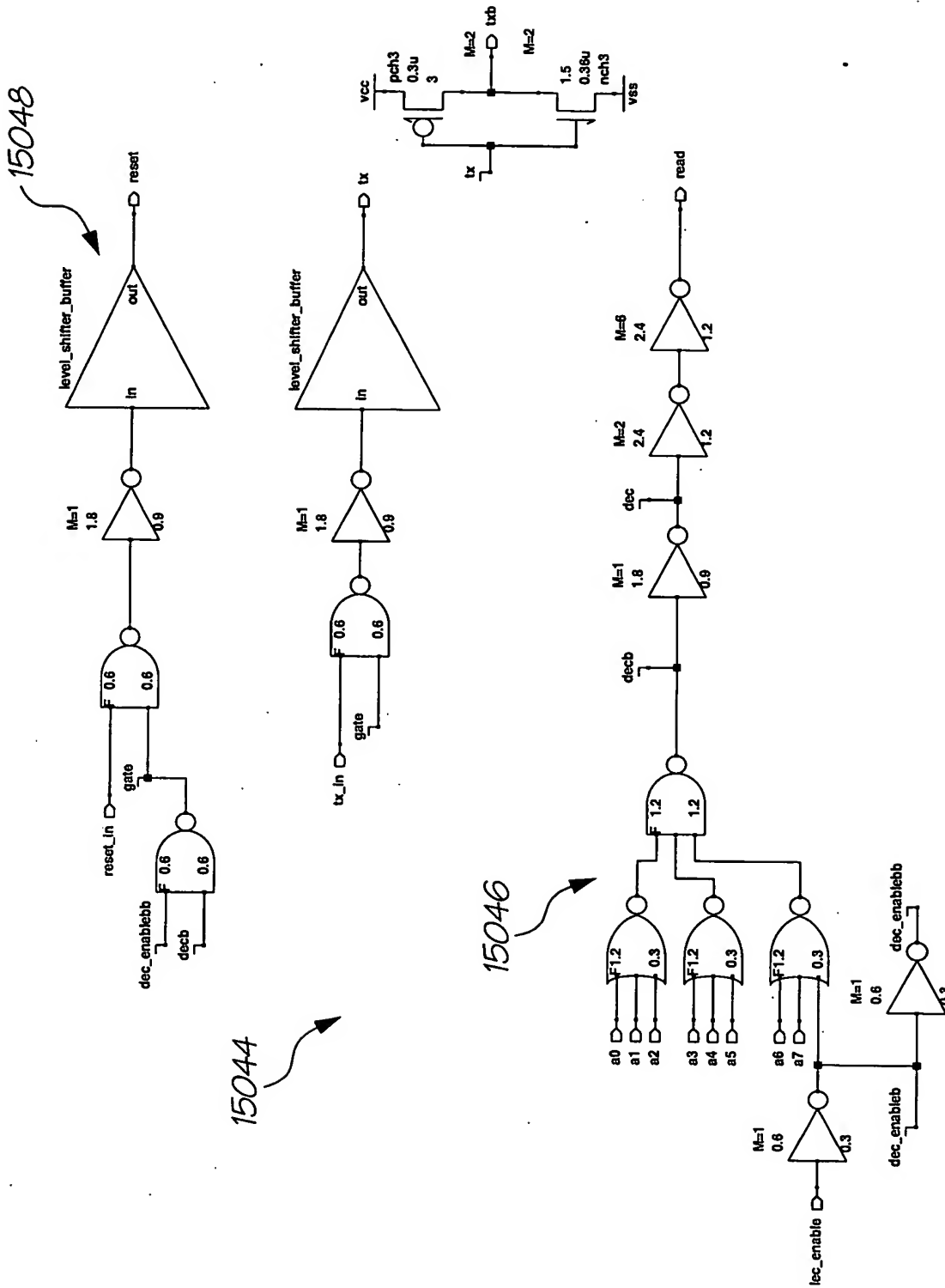






FIG. 115

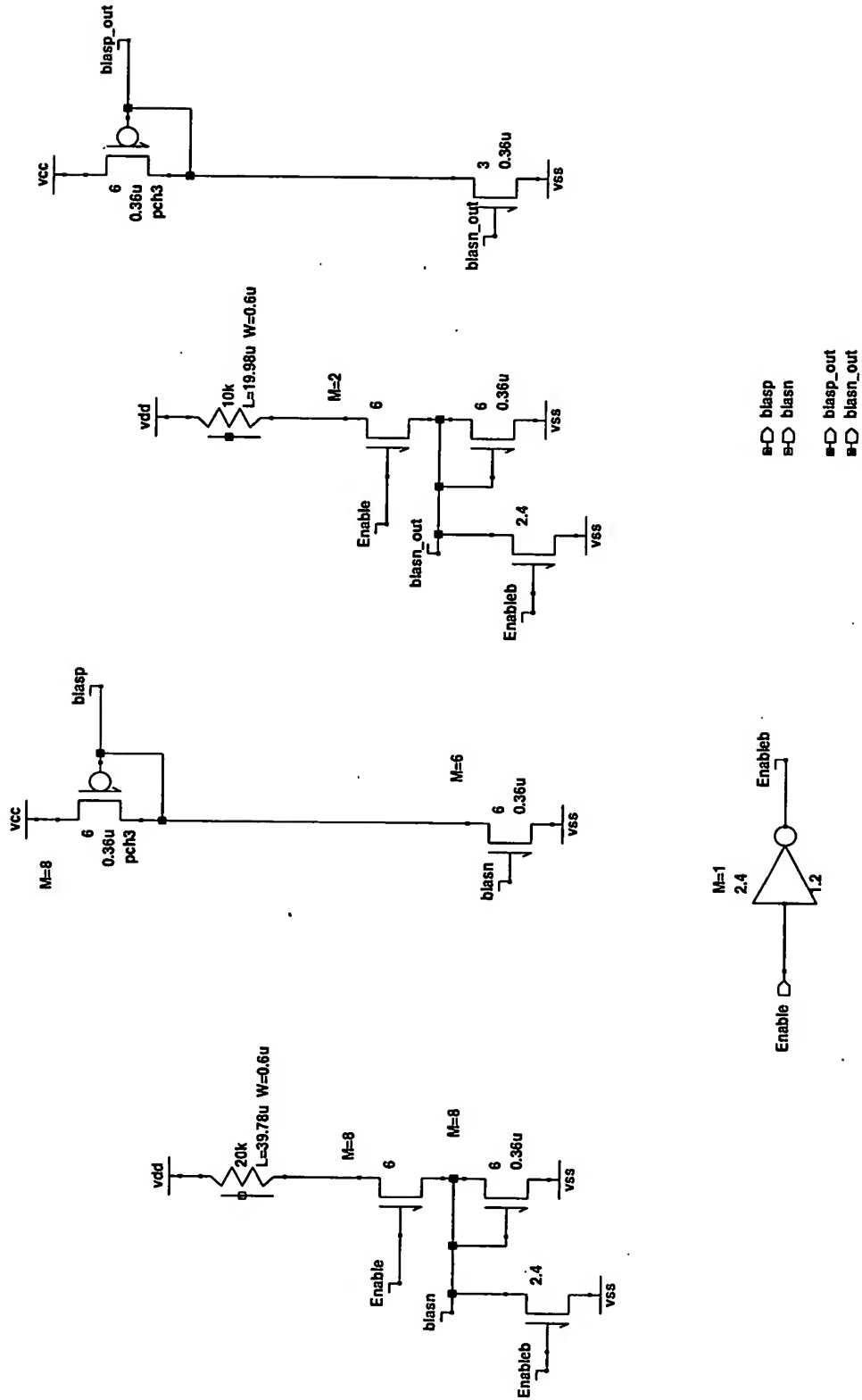


FIG. 116

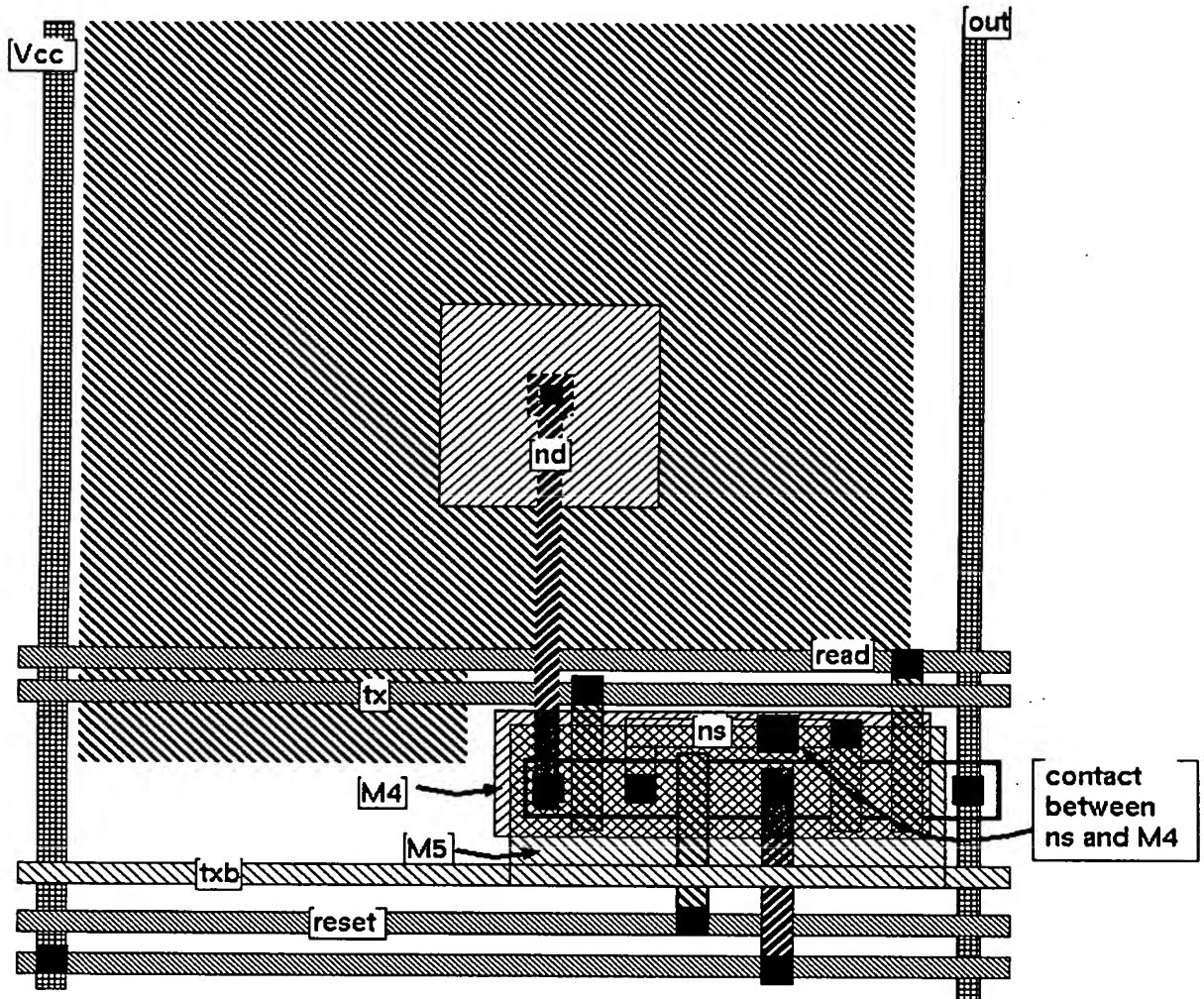


FIG. 117

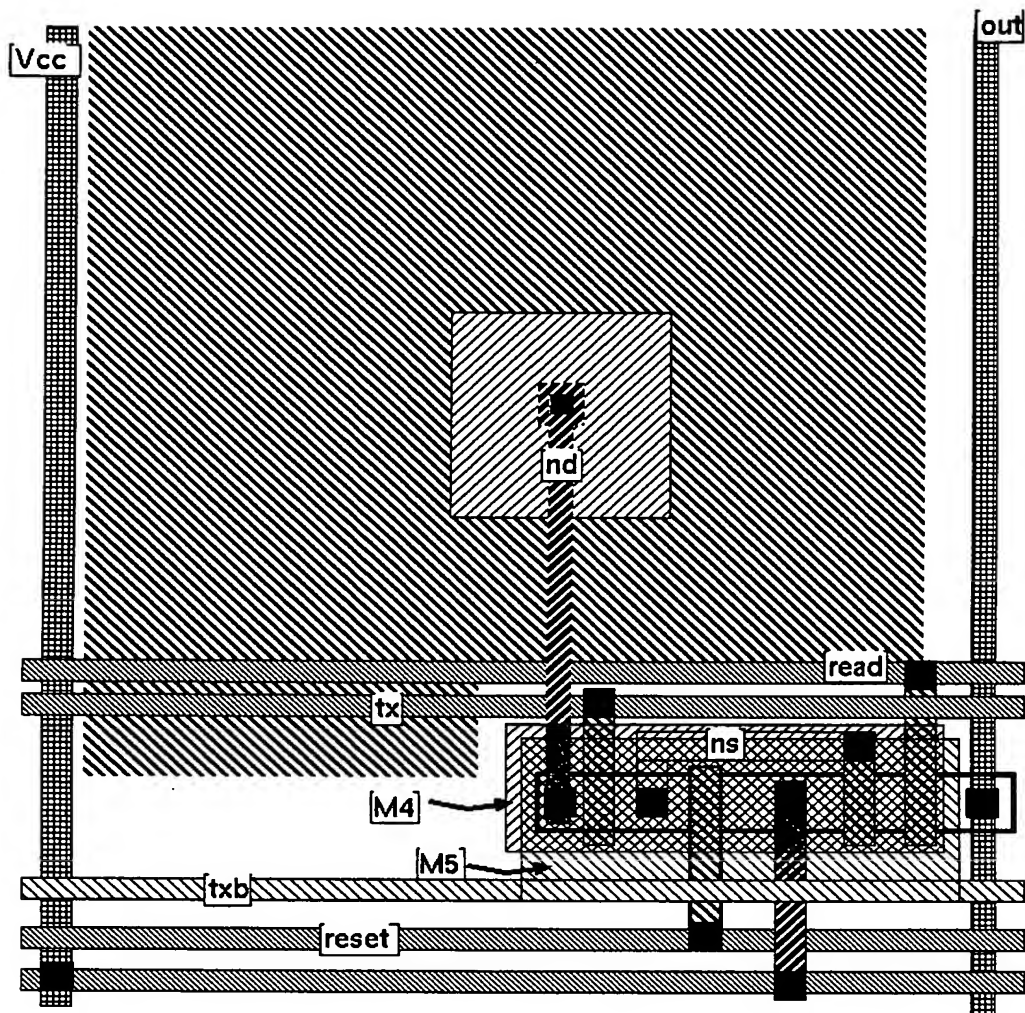


FIG. 118

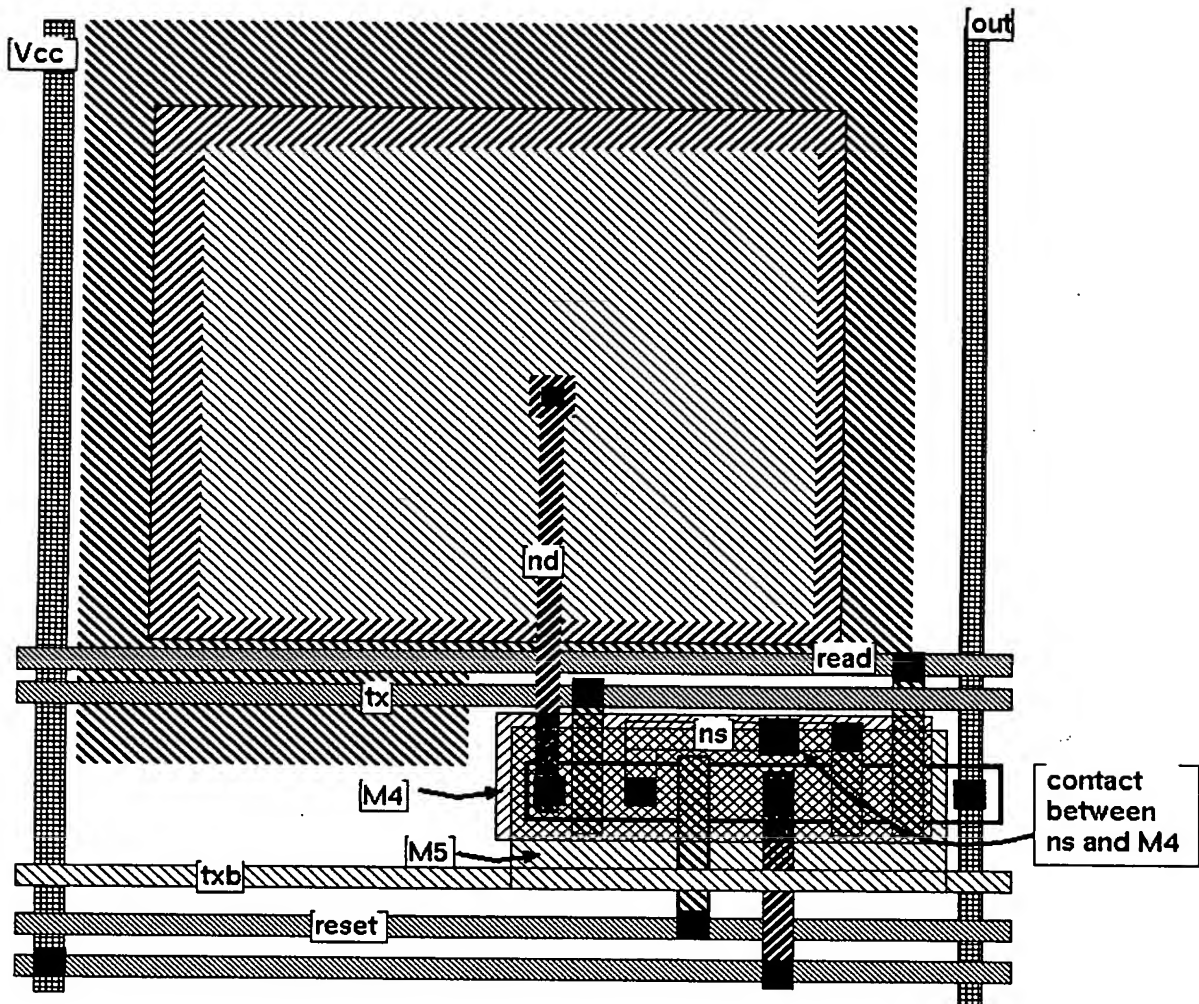


FIG. 119



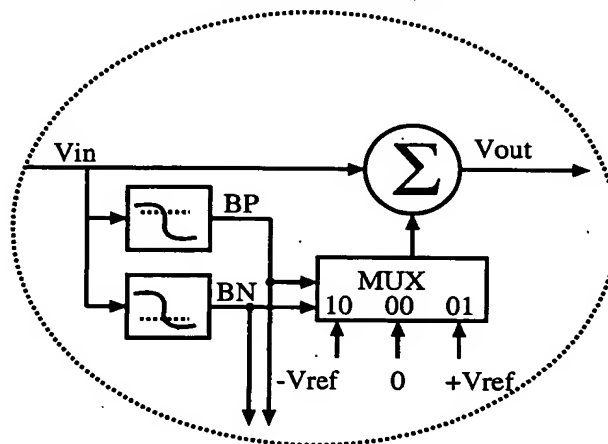
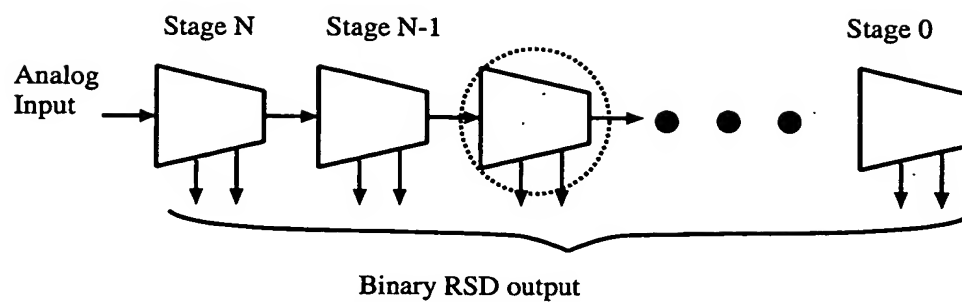


FIG. 121

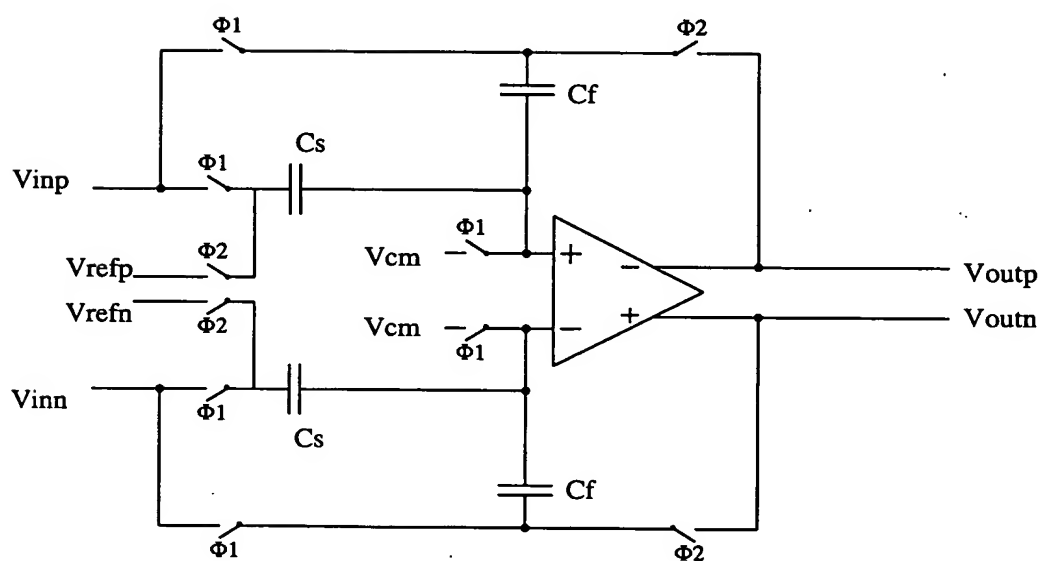


FIG. 122

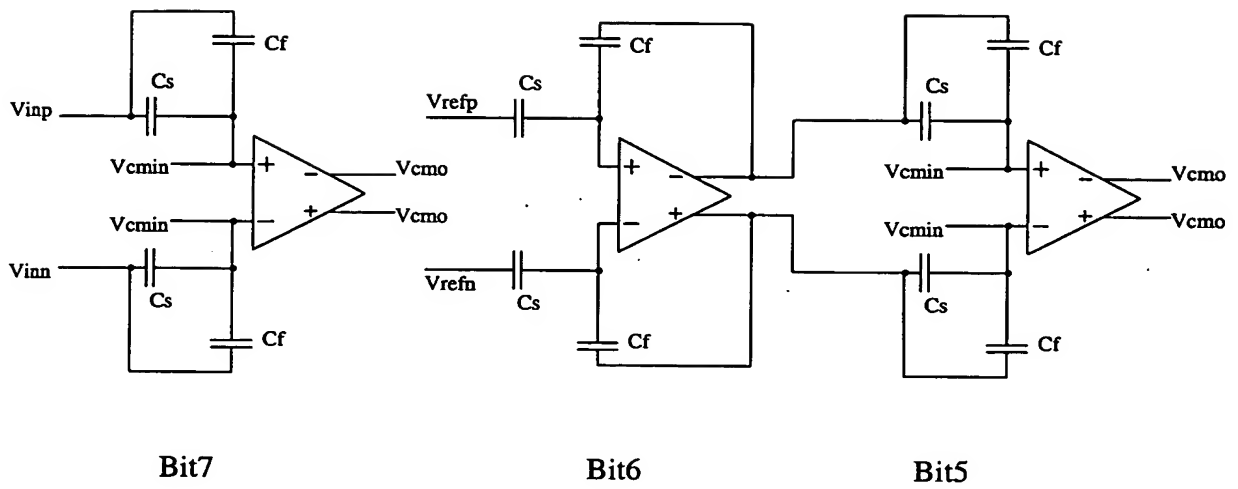


FIG. 123

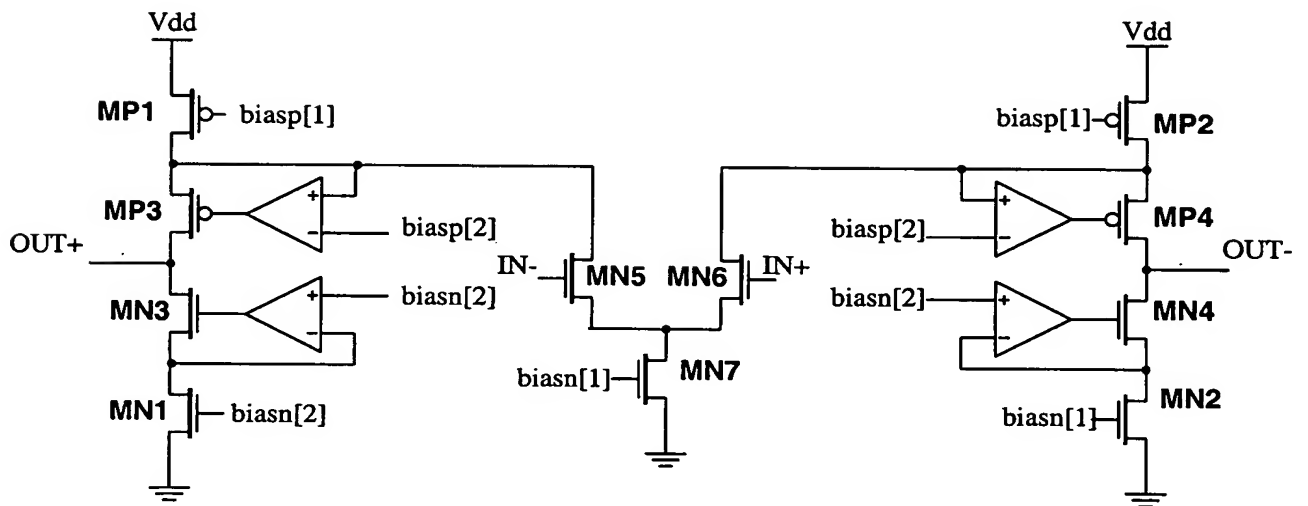


FIG. 124



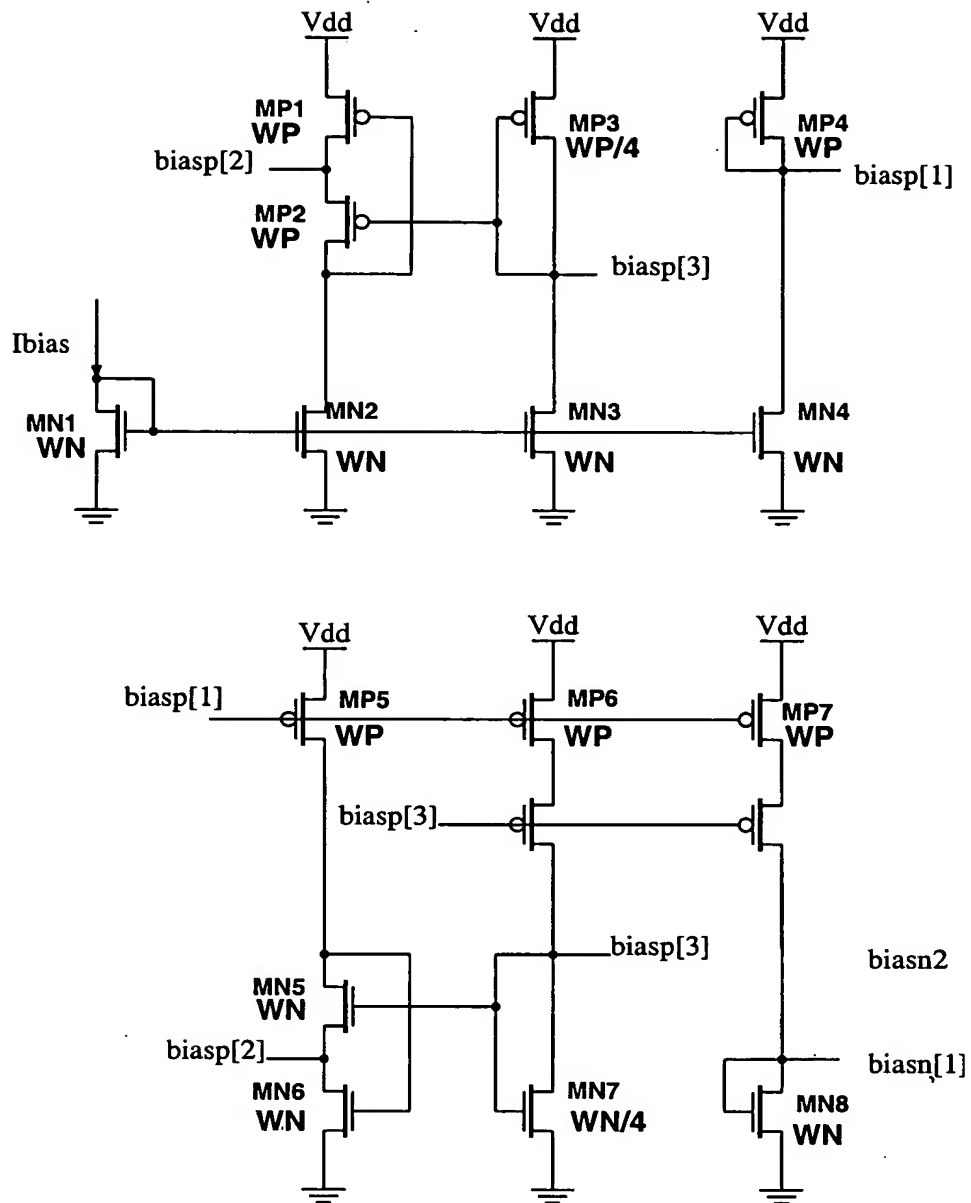


FIG. 125

FIG. 126

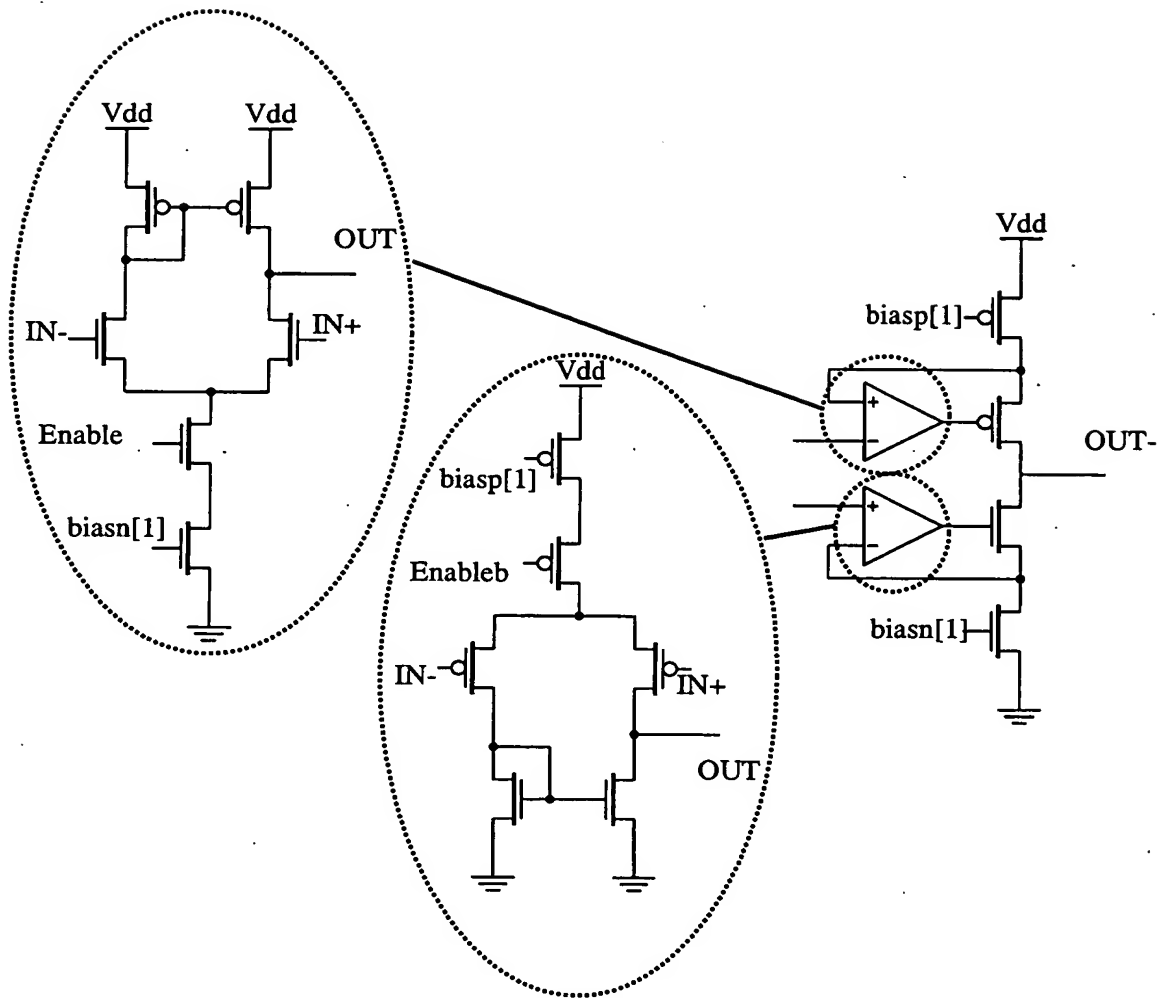


FIG. 127

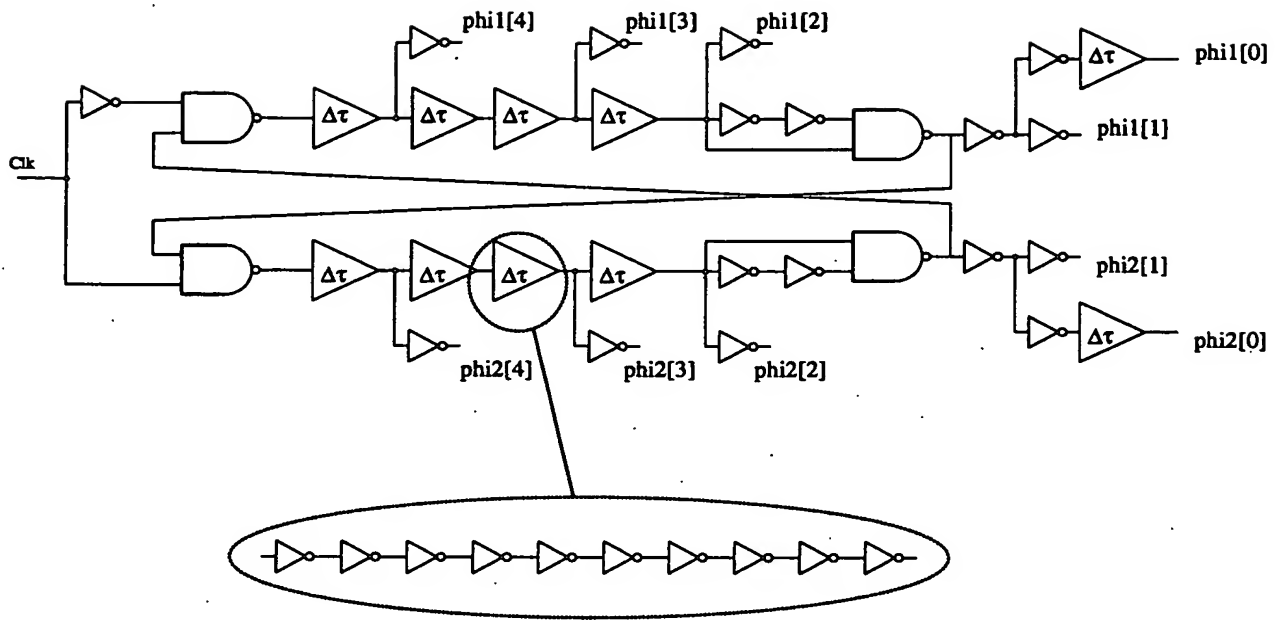


FIG. 128

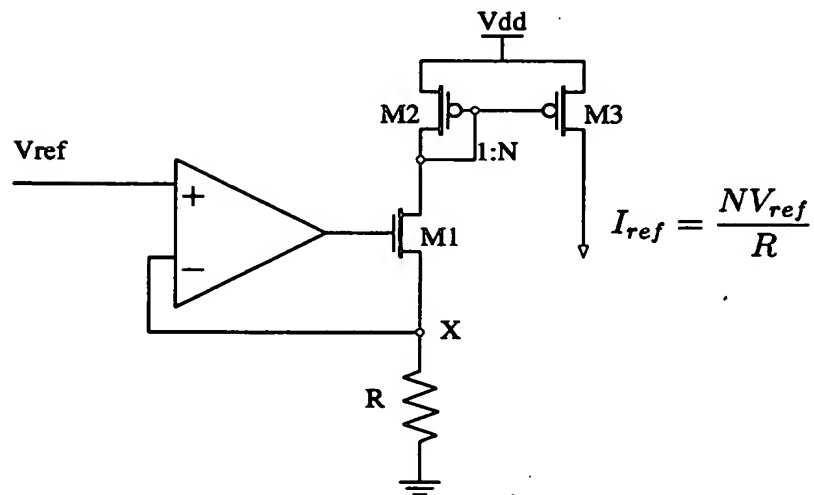


FIG. 129

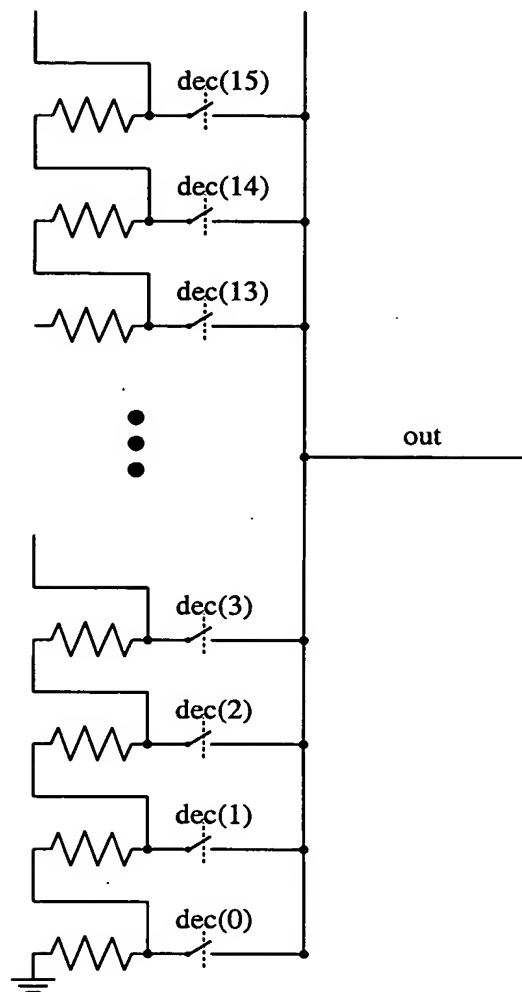


FIG. 130

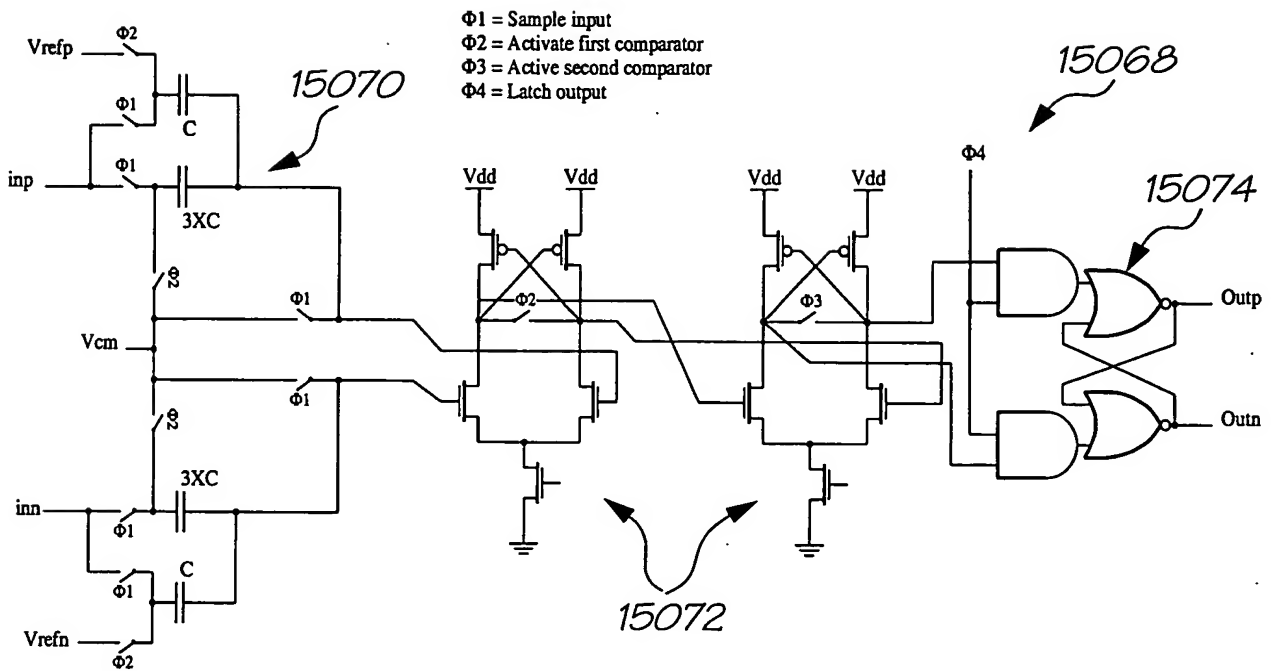


FIG. 131

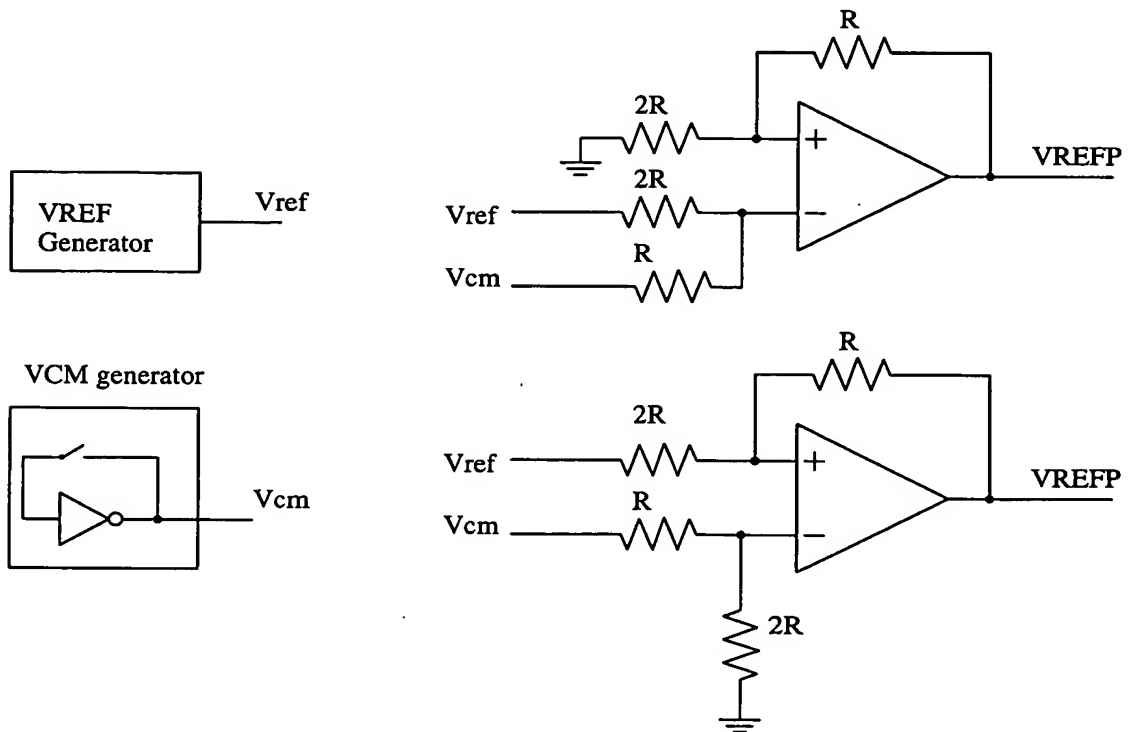


FIG. 132

107/157

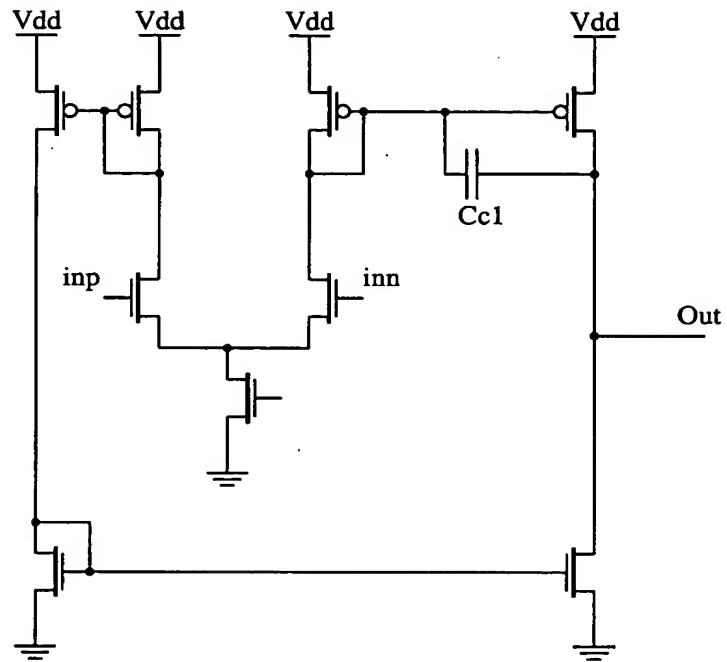


FIG. 133

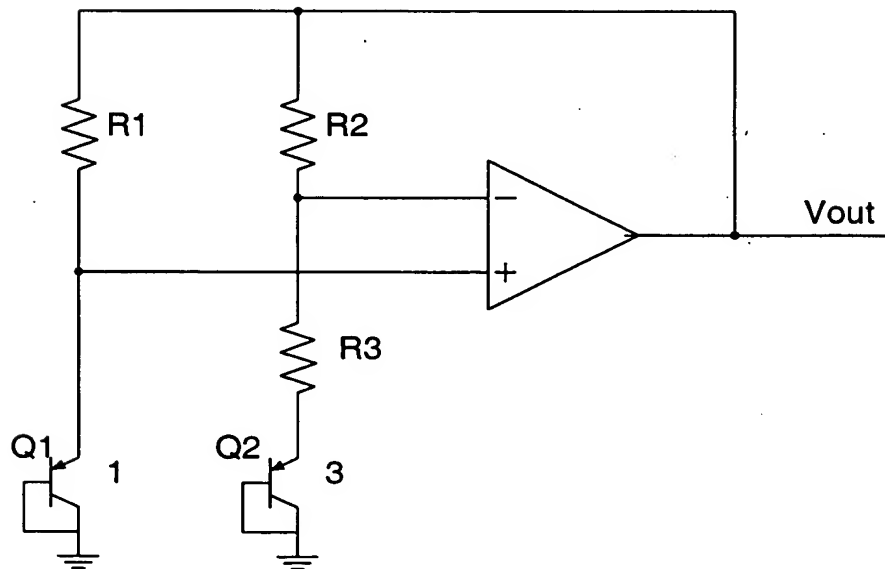


FIG. 134

108/157

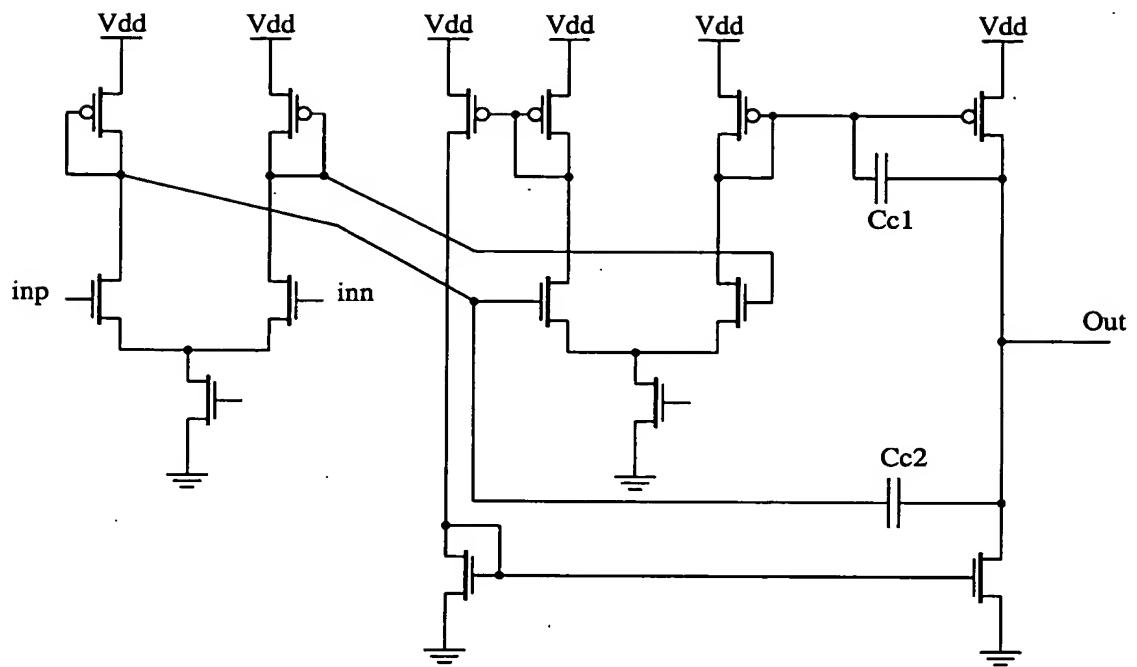


FIG. 135

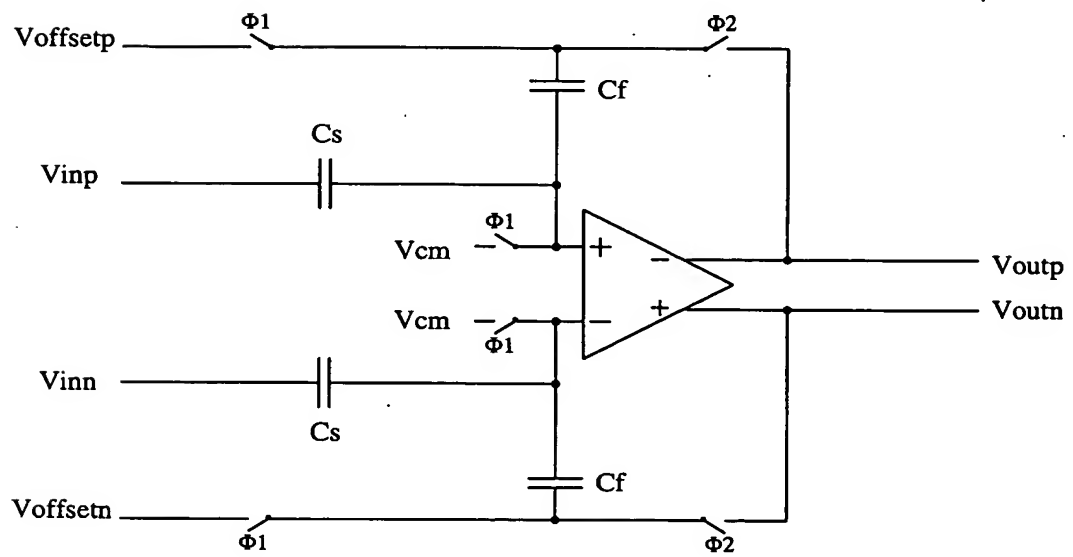


FIG. 136



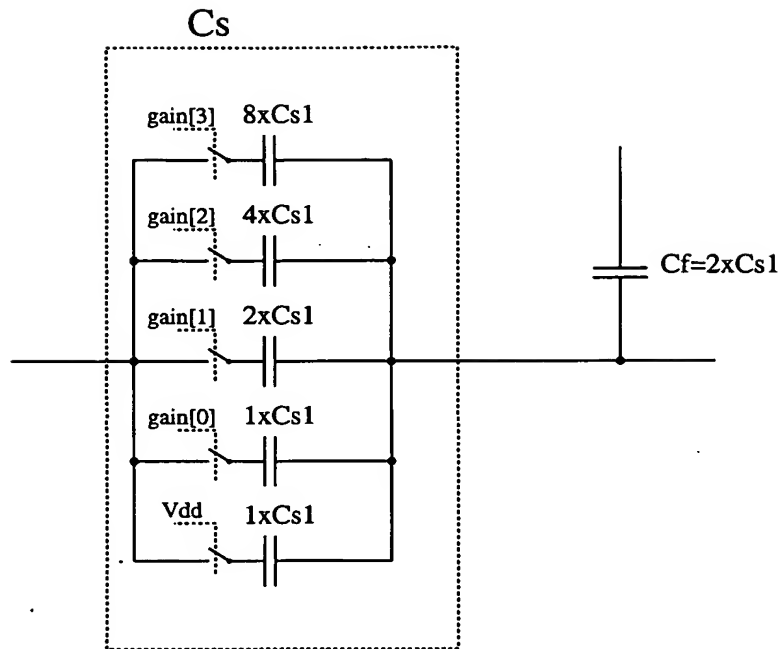


FIG. 137

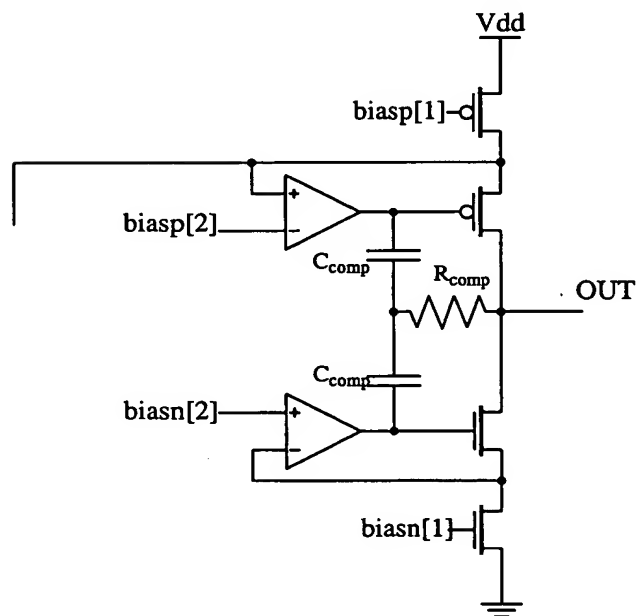


FIG. 138

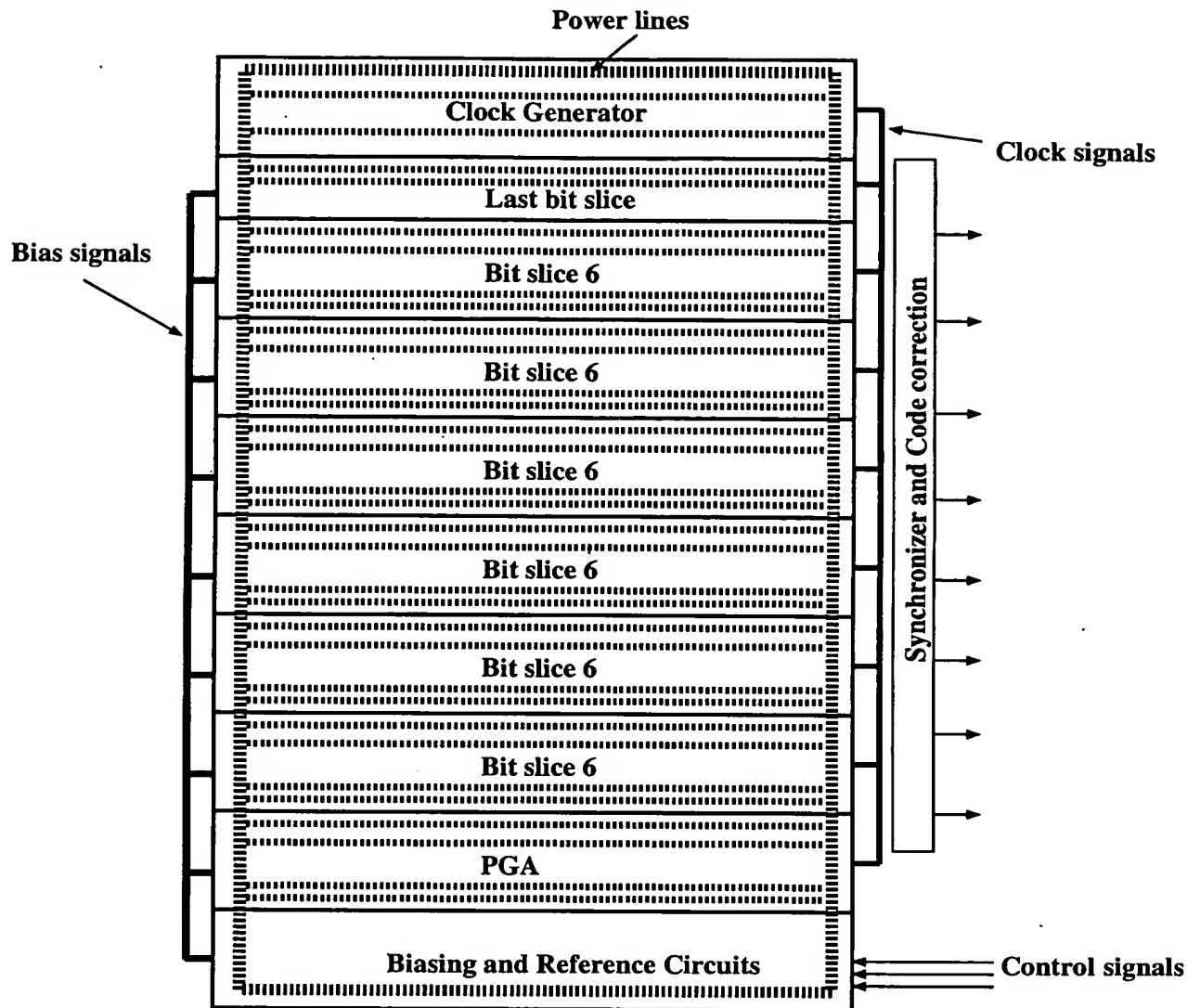


FIG. 139

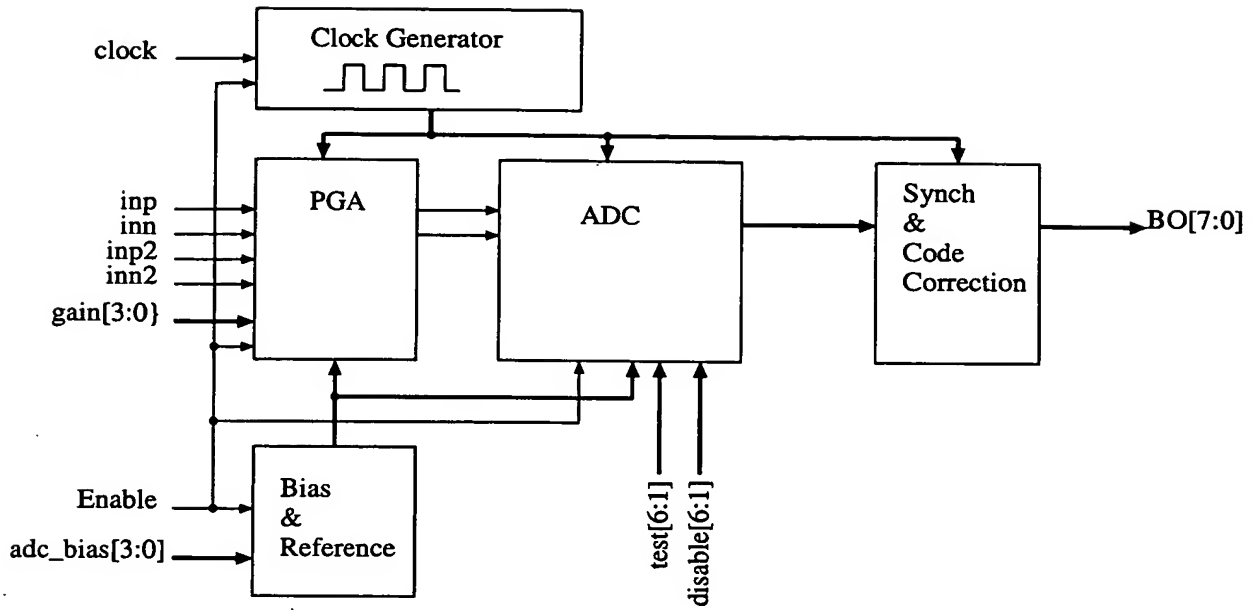


FIG. 140

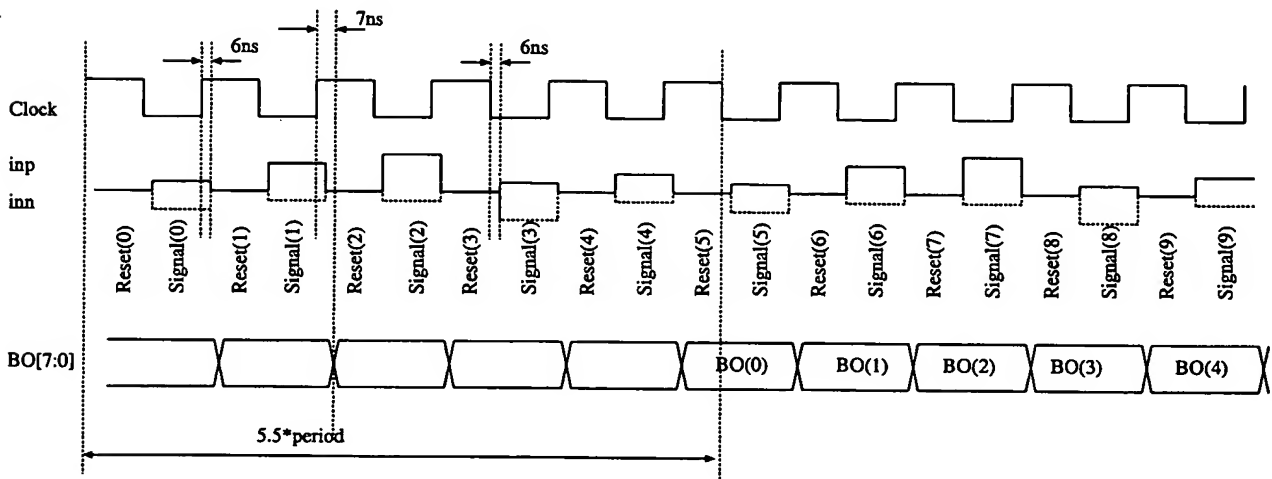


FIG. 141

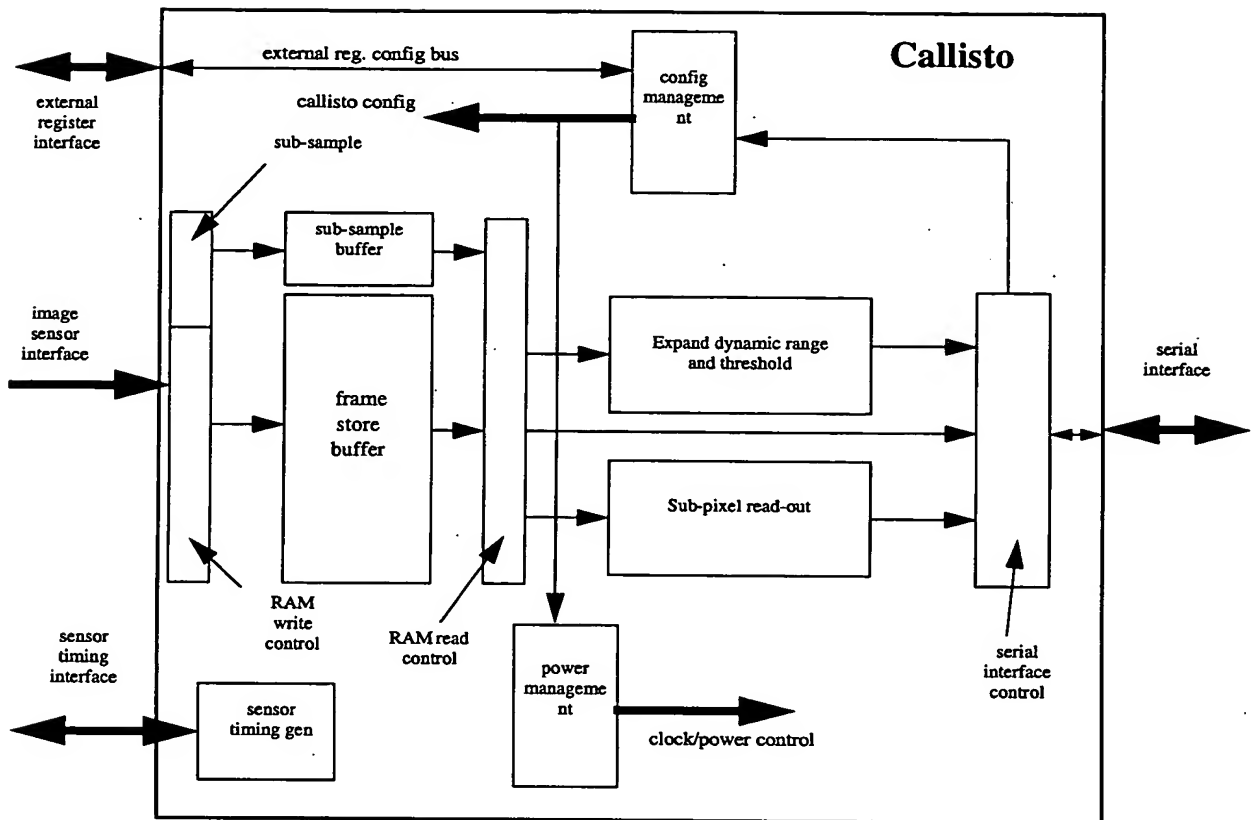


FIG. 142

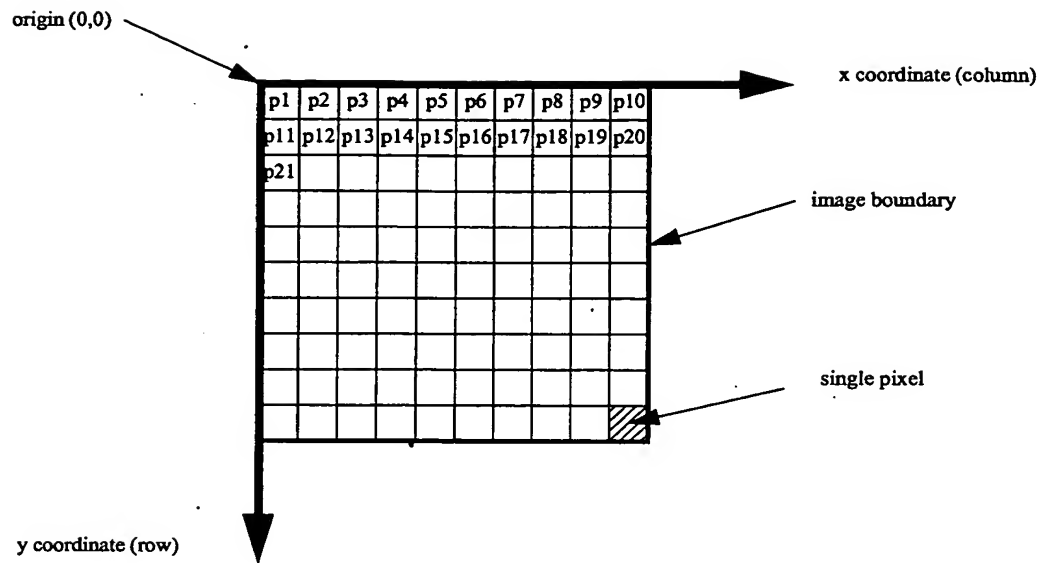


FIG. 143

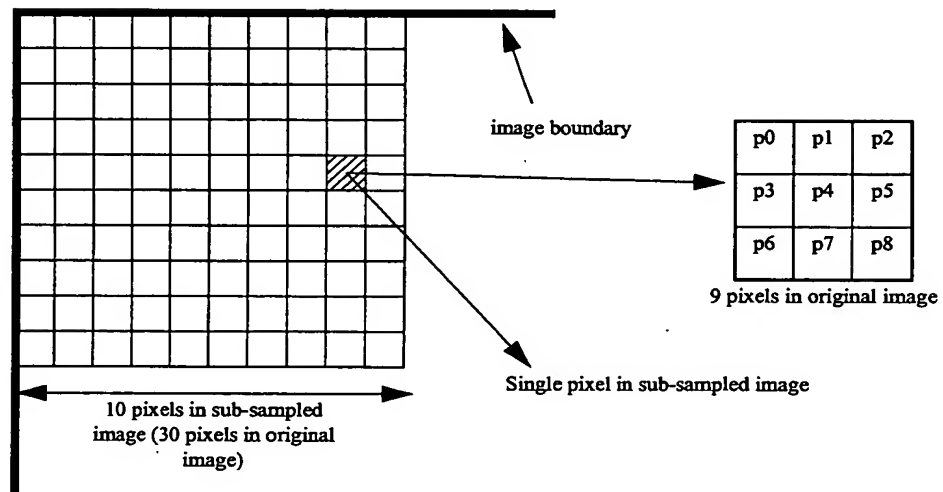


FIG. 144

114/157

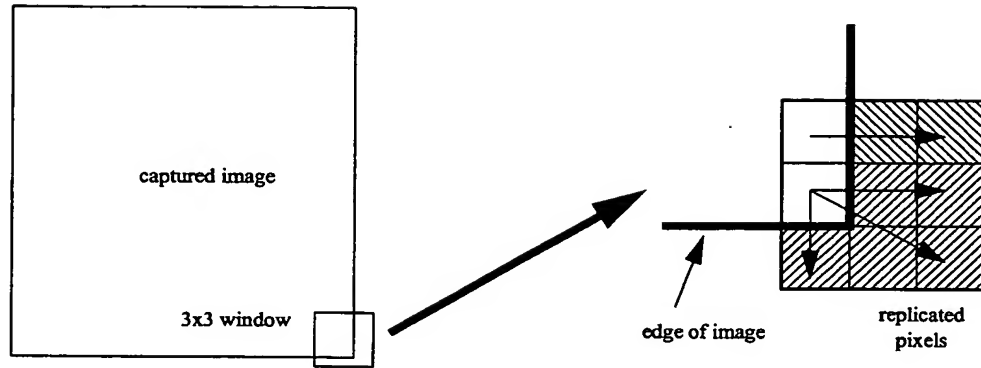


FIG. 145

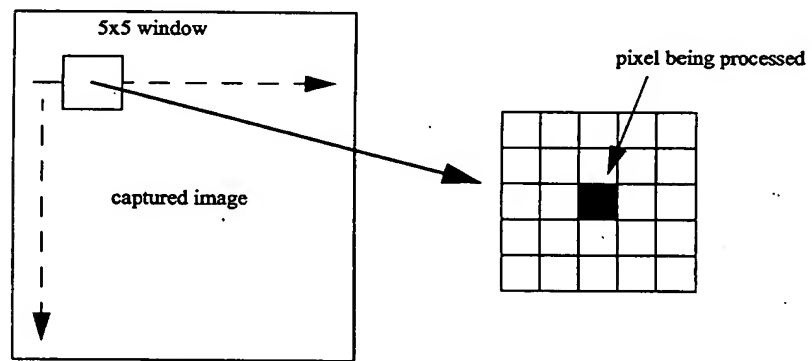


FIG. 146

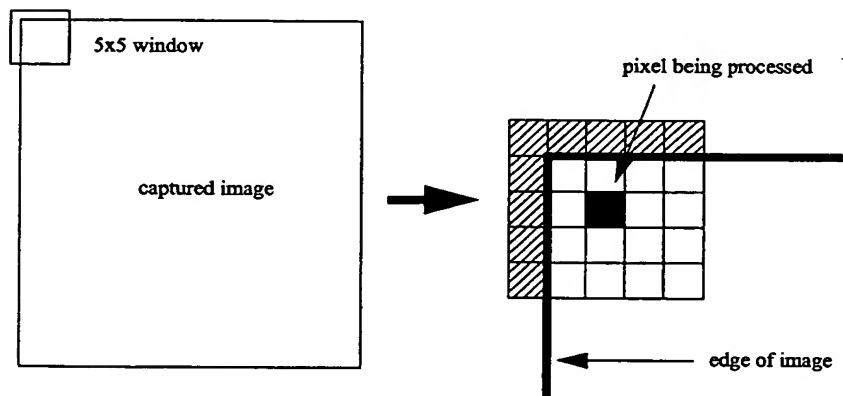


FIG. 147

115/157

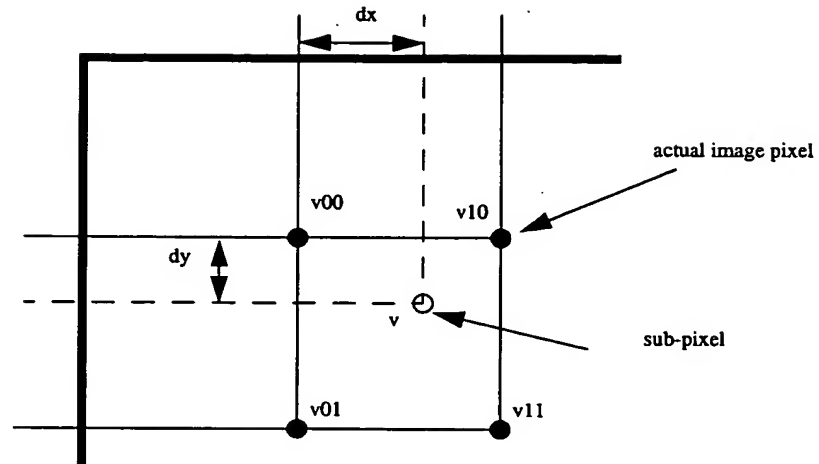


FIG. 148

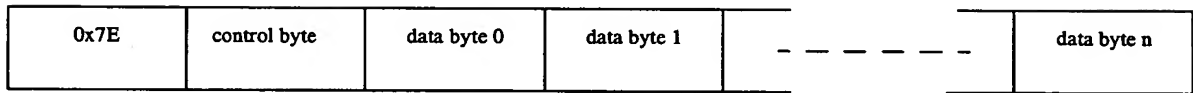


FIG. 149

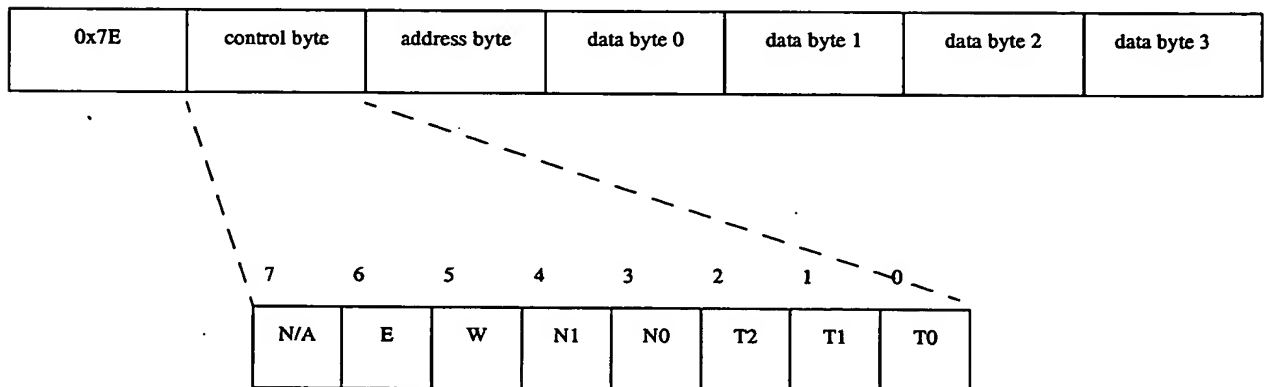


FIG. 150

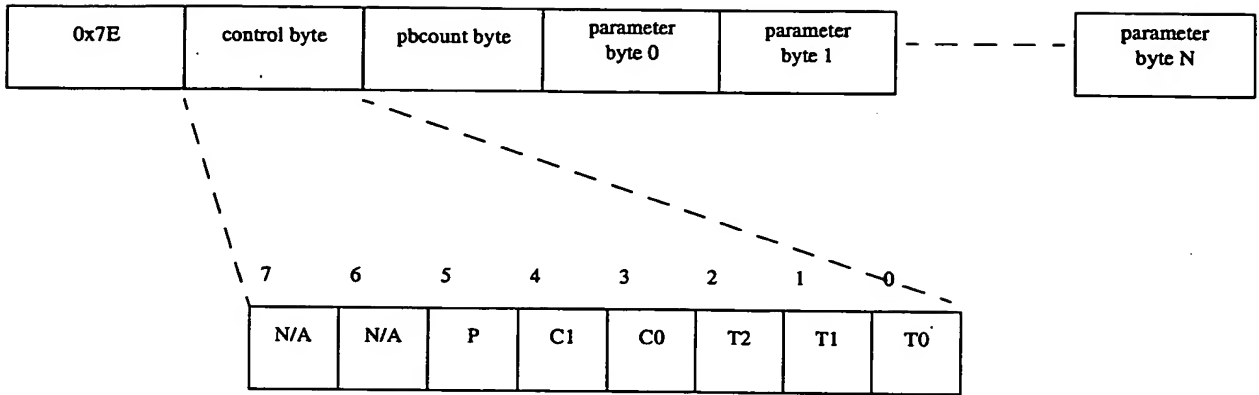


FIG. 151

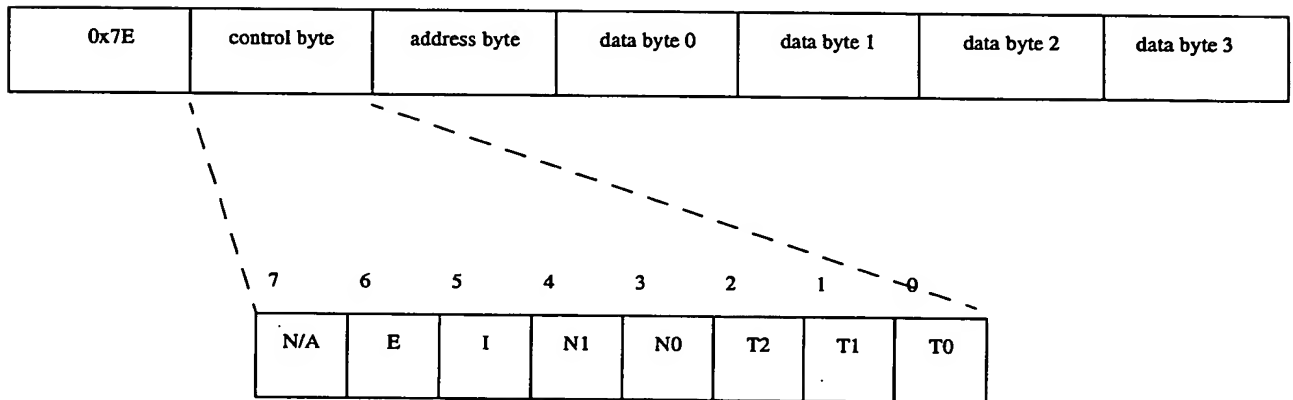


FIG. 152



117/157

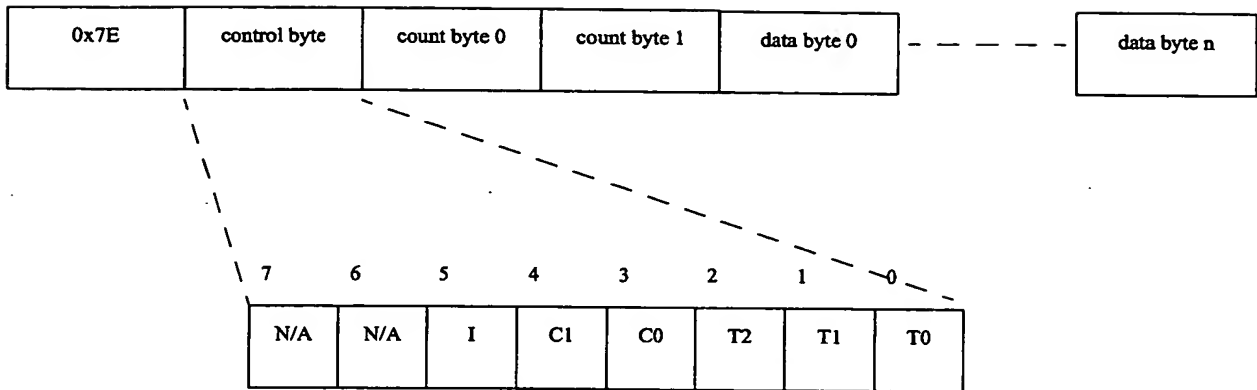


FIG. 153

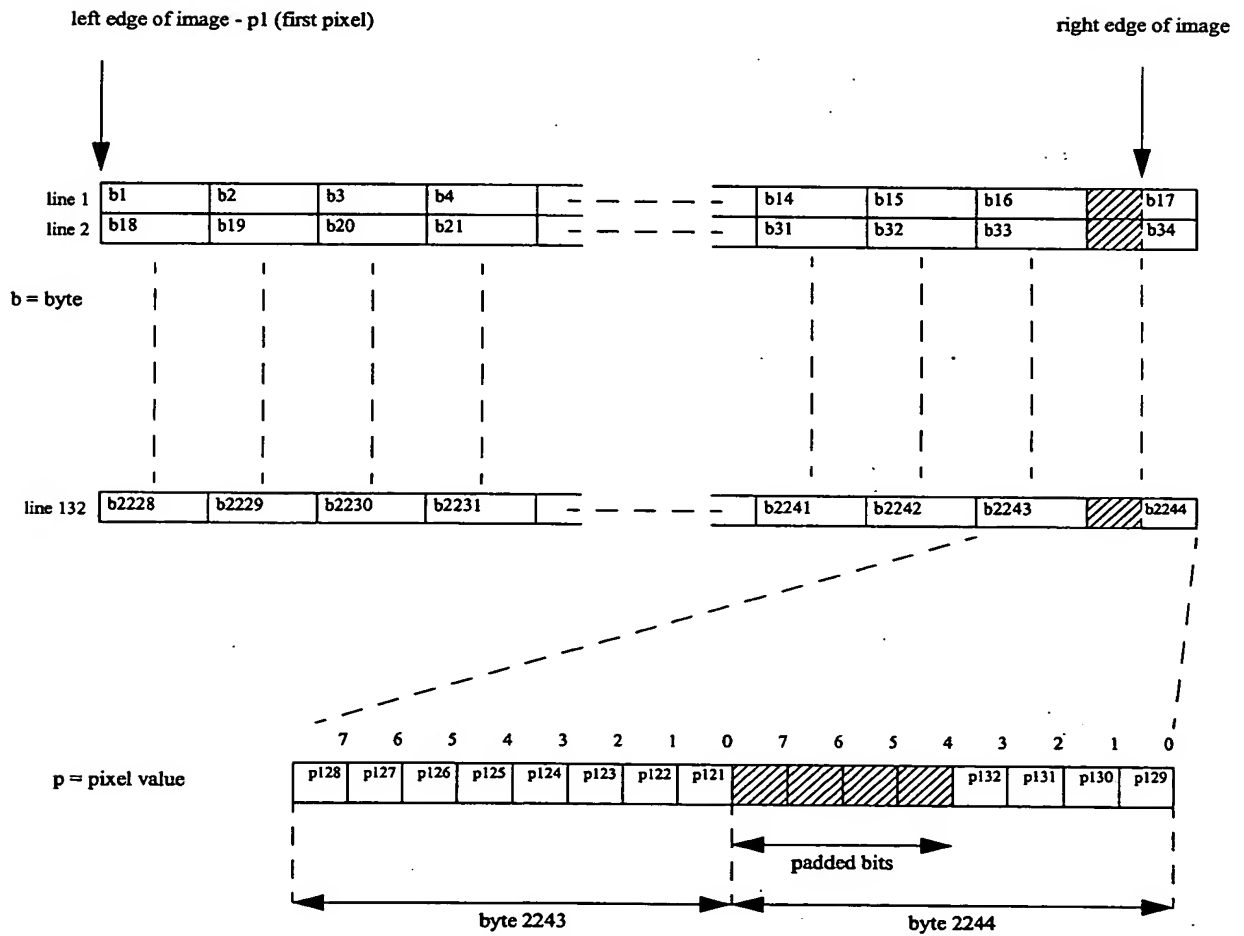


FIG. 154

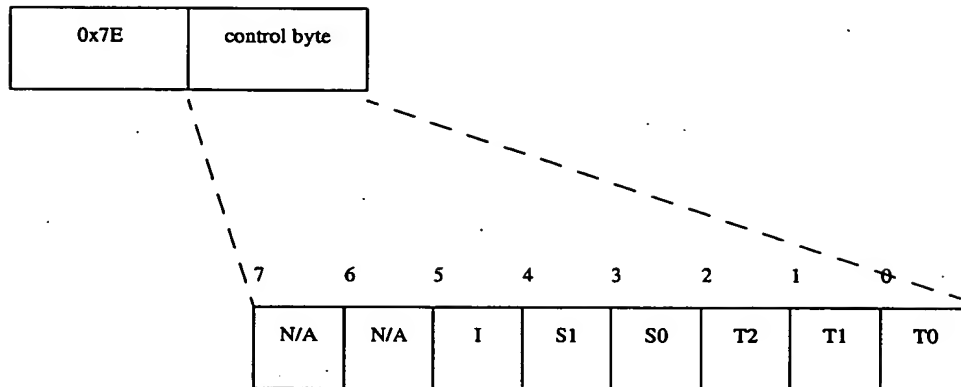


FIG. 155

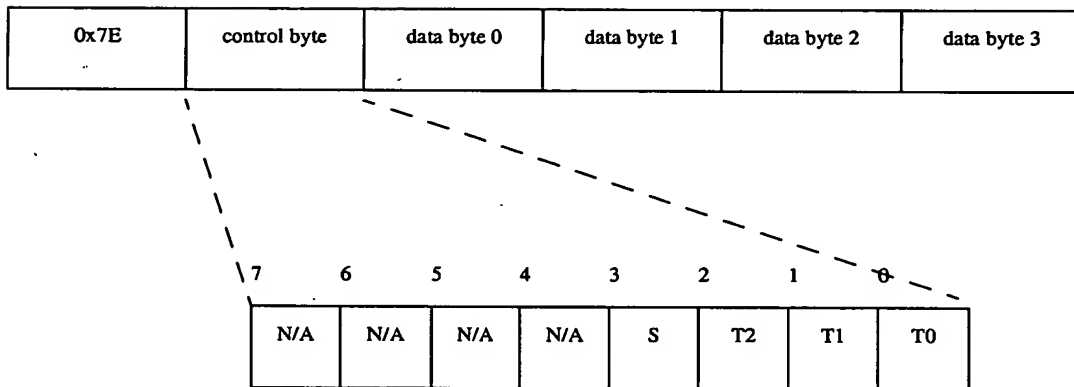


FIG. 156

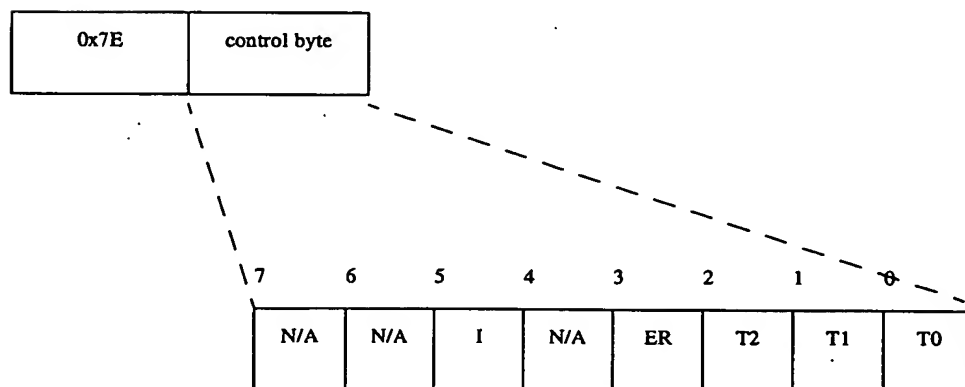


FIG. 157

119/157

0x7E	control byte "00100001"	TopLeftX	TopLeftY
------	----------------------------	----------	----------

Unprocessed image read command

*FIG. 158*

0x7E	control byte "00111001"	TopLeftX	TopLeftY	SizeX-1	SizeY-1
------	----------------------------	----------	----------	---------	---------

processed image read command with arguments

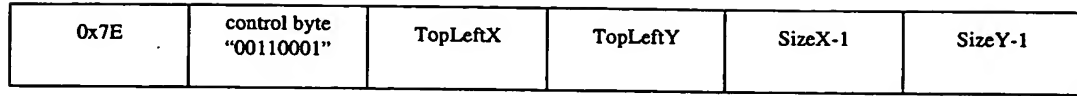
*FIG. 159a*

0x7E	control byte "00011001"
------	----------------------------

processed image read command without arguments

*FIG. 159b*

120/157



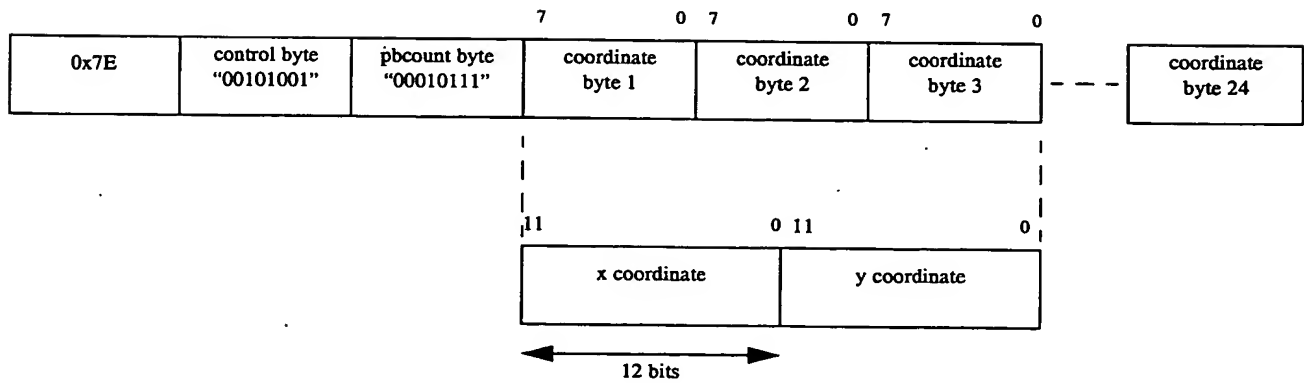
Sub-sampled image read command with arguments

*FIG. 160a*



Sub-sampled image read command without arguments

*FIG. 160b*



*FIG. 161*

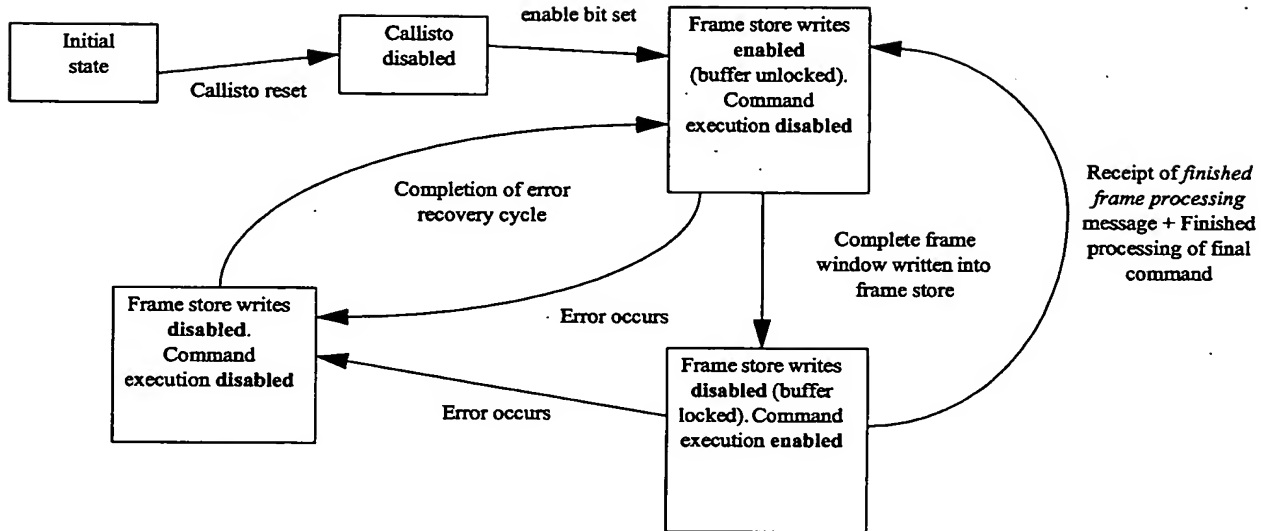


FIG. 162

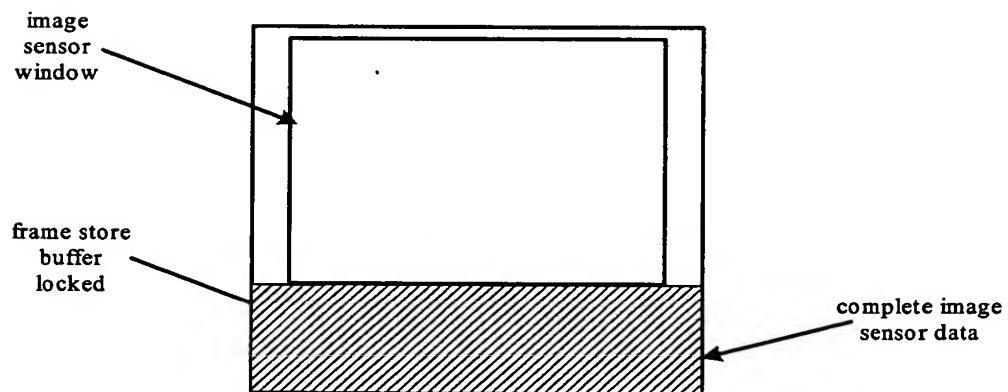


FIG. 163

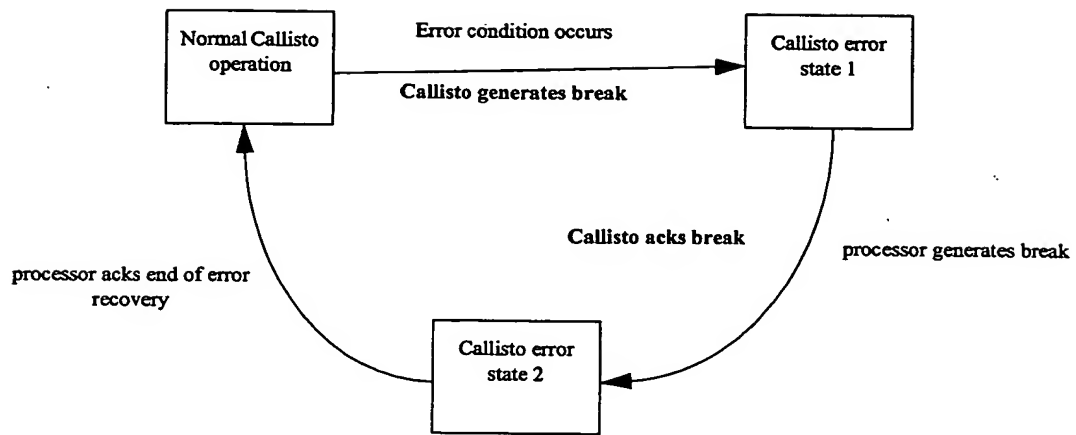


FIG. 164

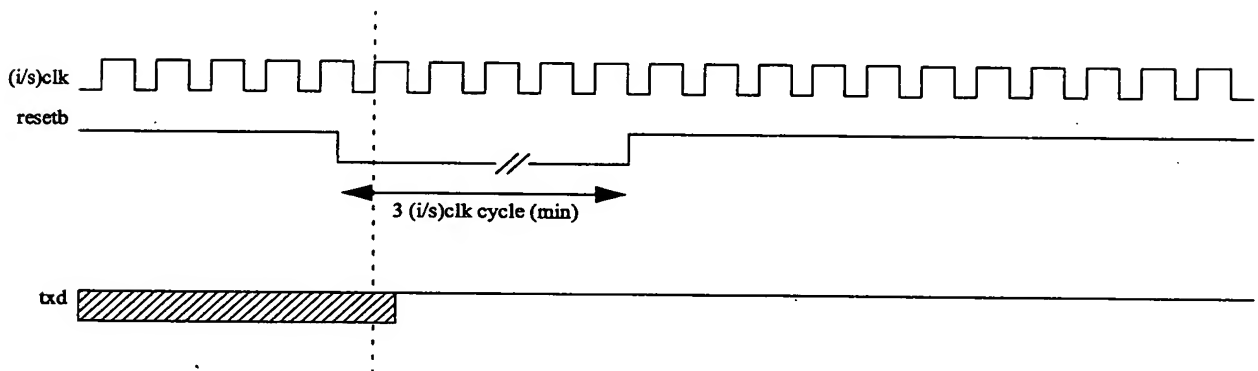


FIG. 165

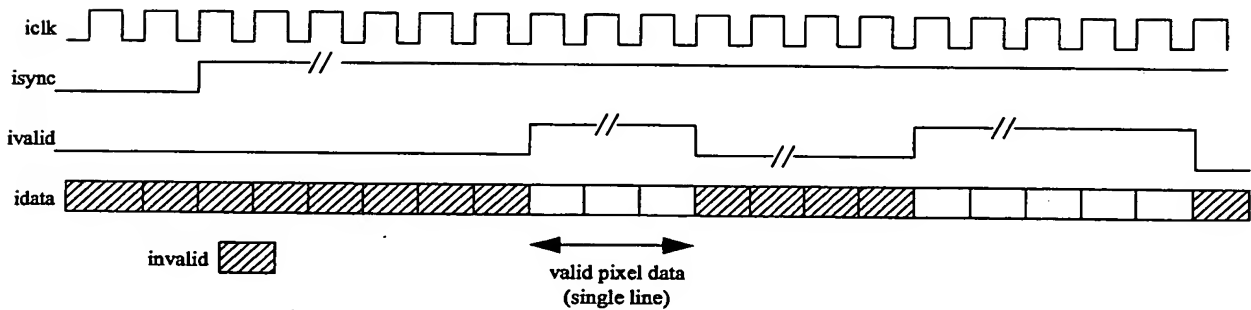


FIG. 166

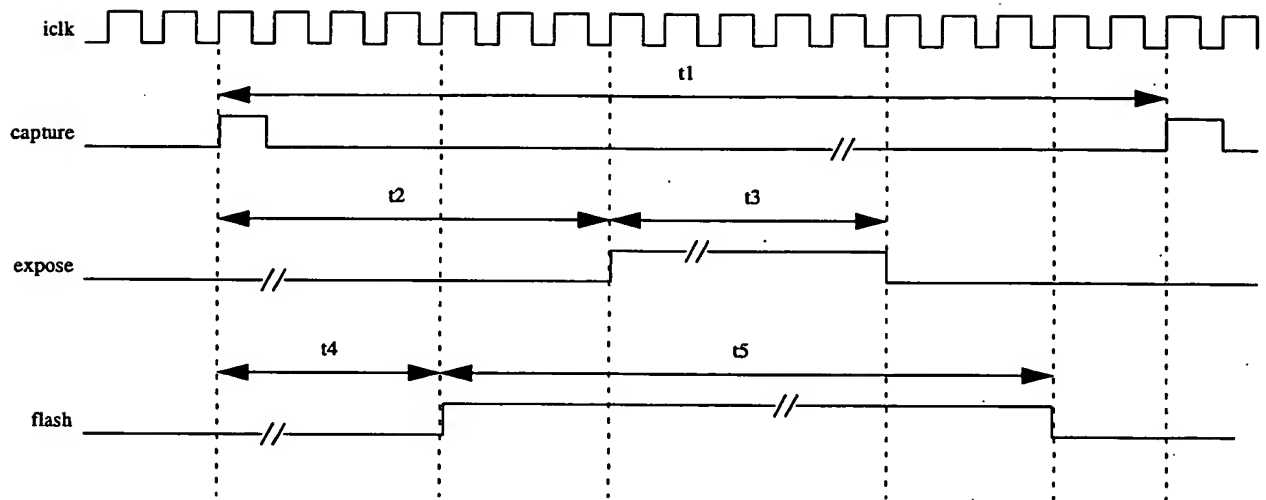


FIG. 167

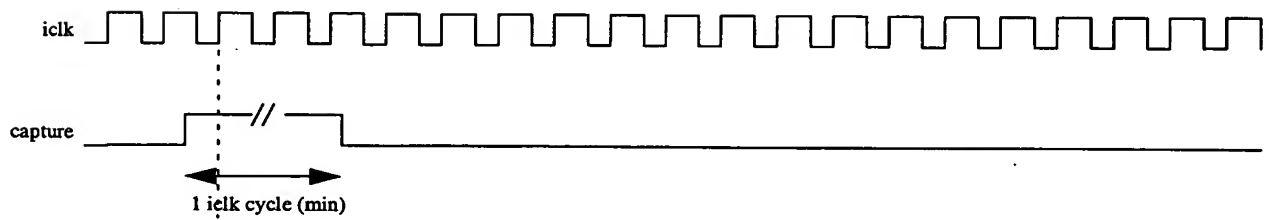


FIG. 168

2 bytes back-to-back from Callisto to microprocessor

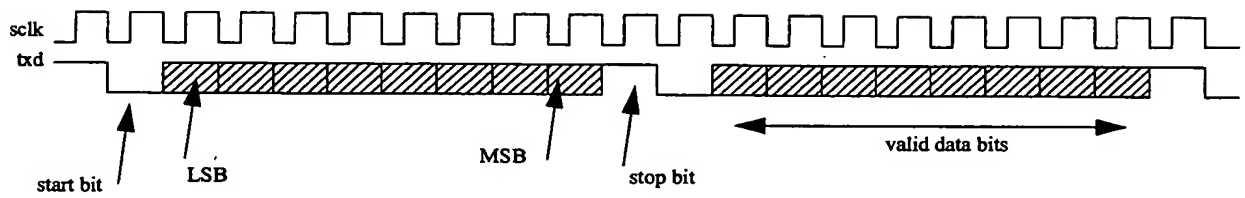


FIG. 169

Single byte transfer from microprocessor to Callisto

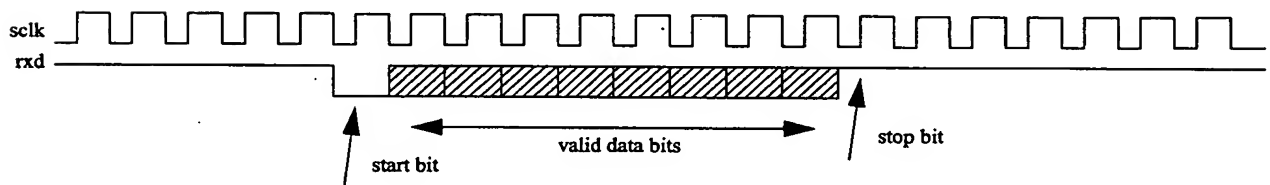


FIG. 170



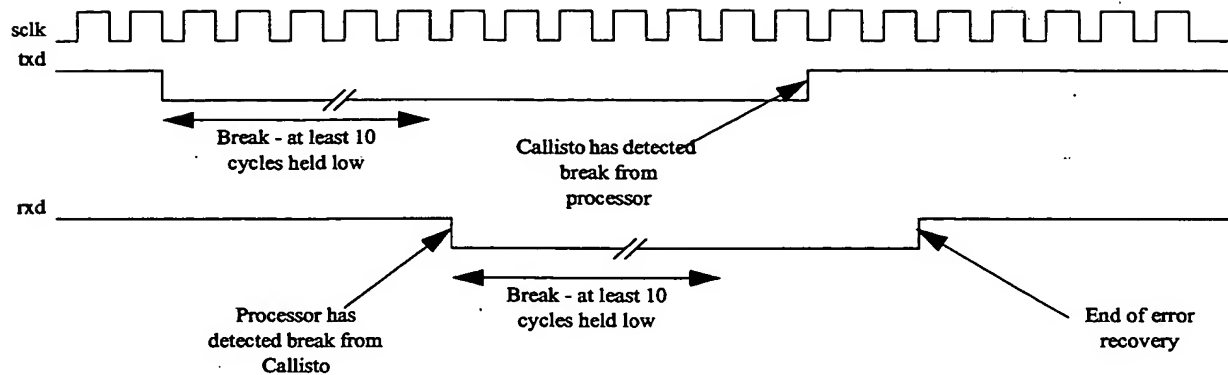


FIG. 171

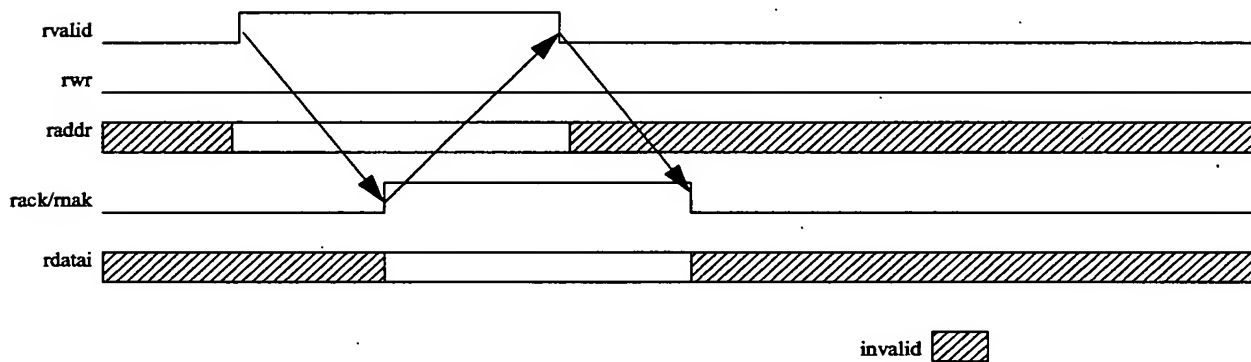


FIG. 172

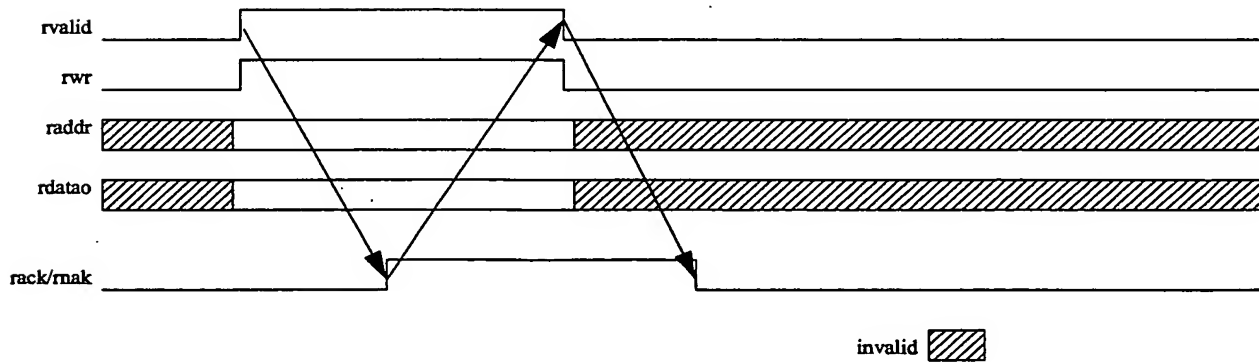


FIG. 173

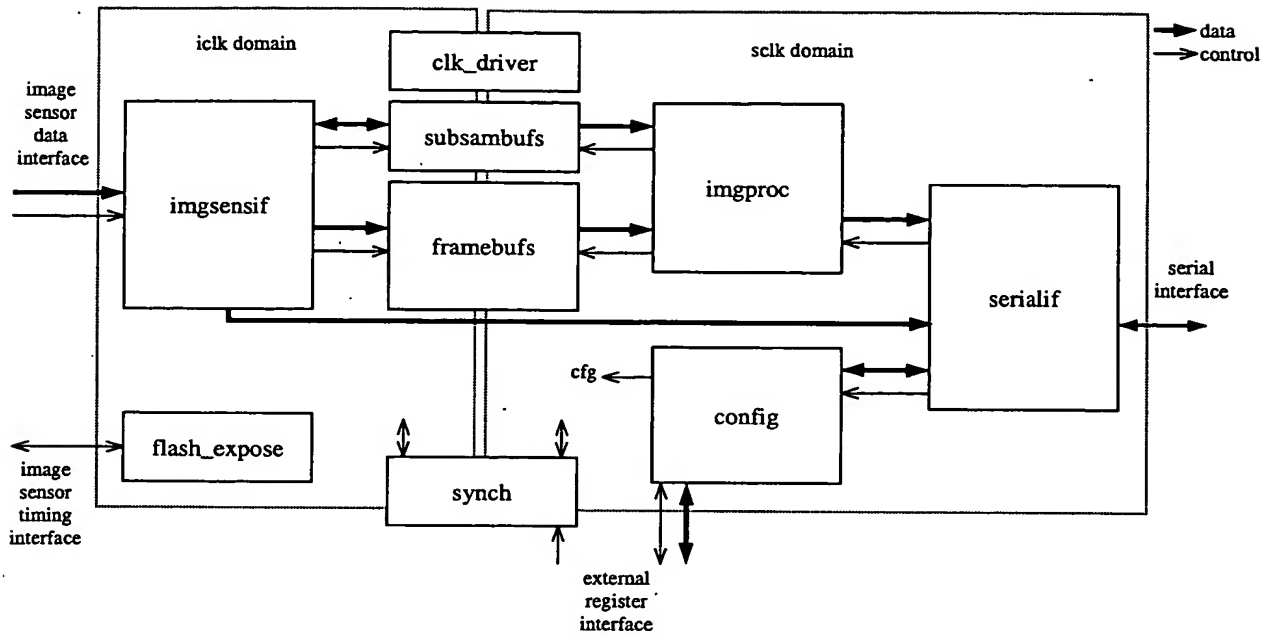


FIG. 174

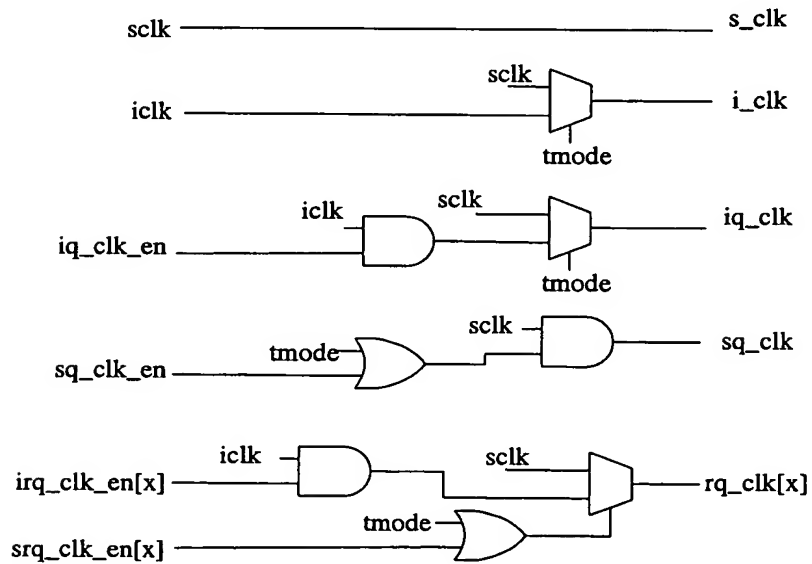


FIG. 175

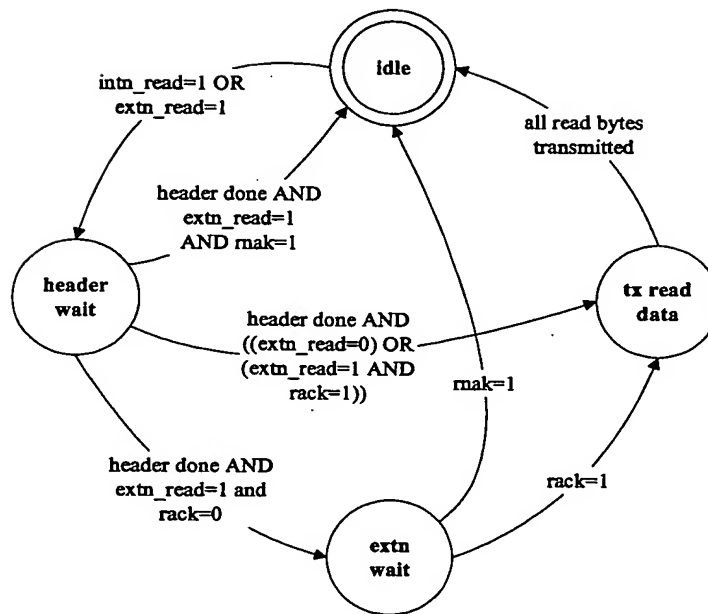


FIG. 176

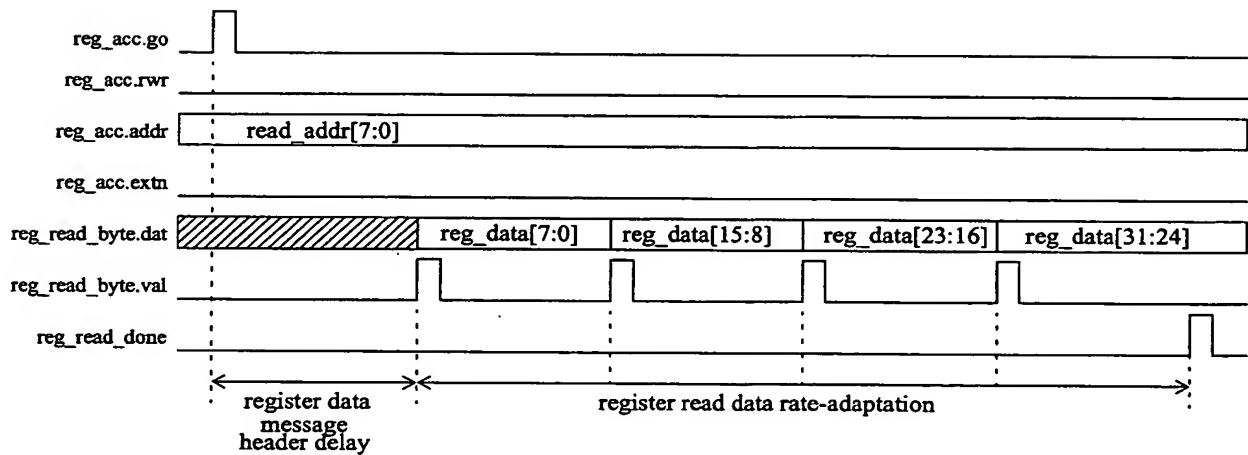


FIG. 176a

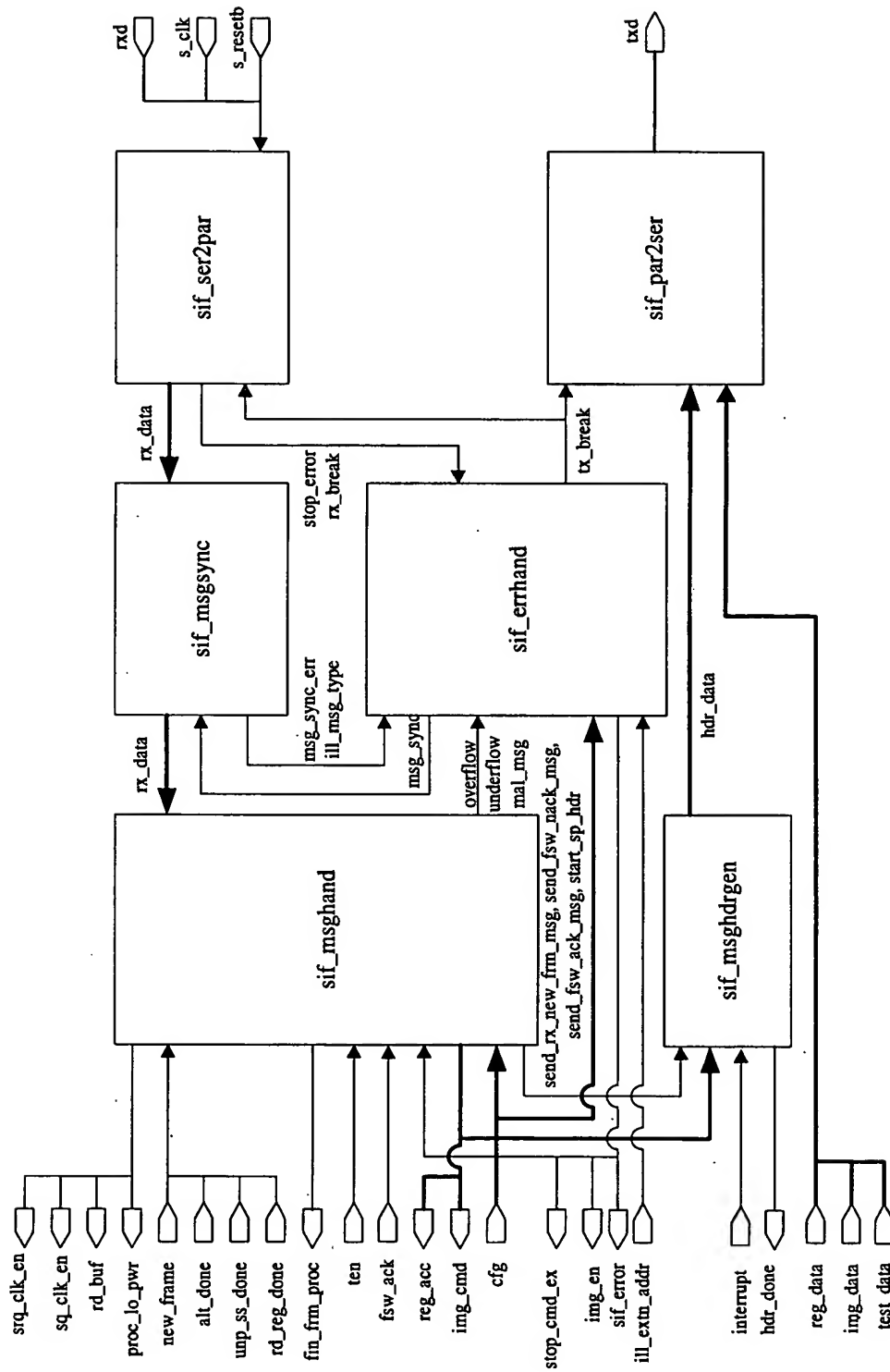


FIG. 177

129/157

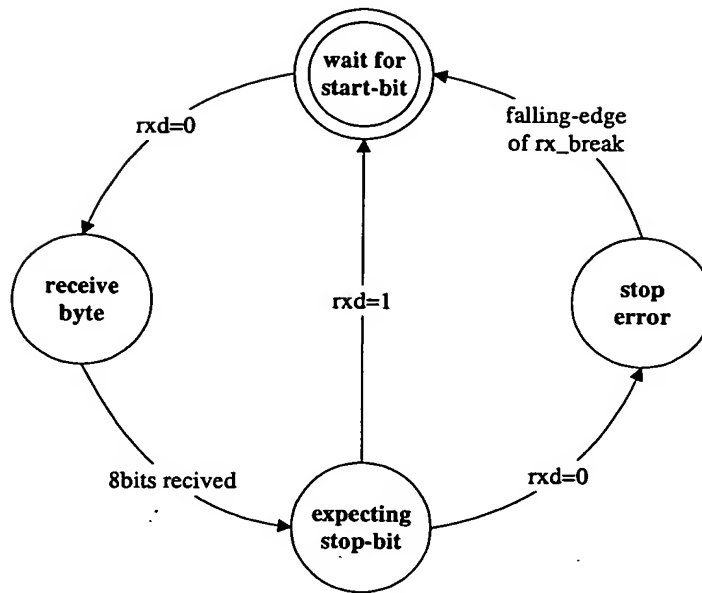


FIG. 178

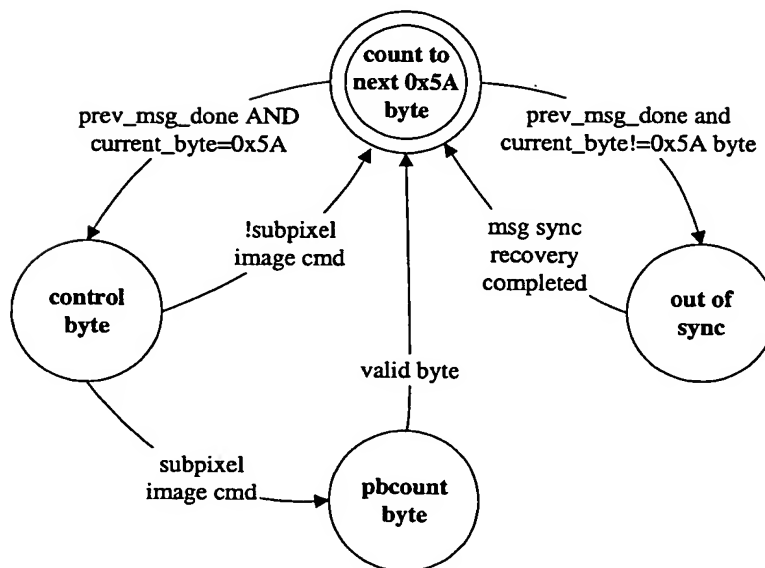


FIG. 179

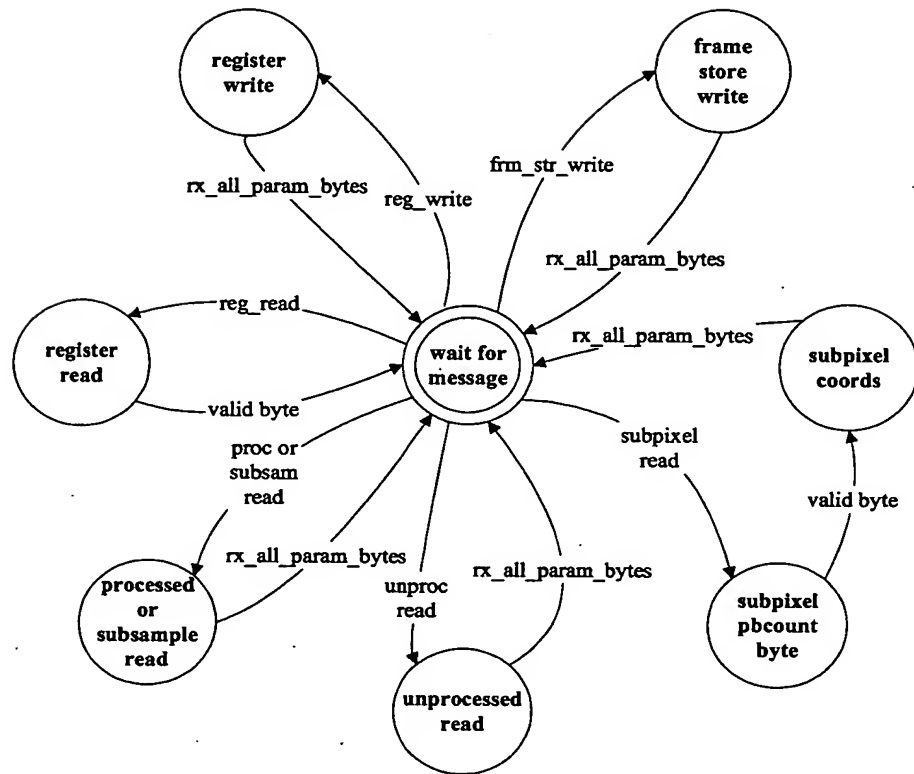


FIG. 180

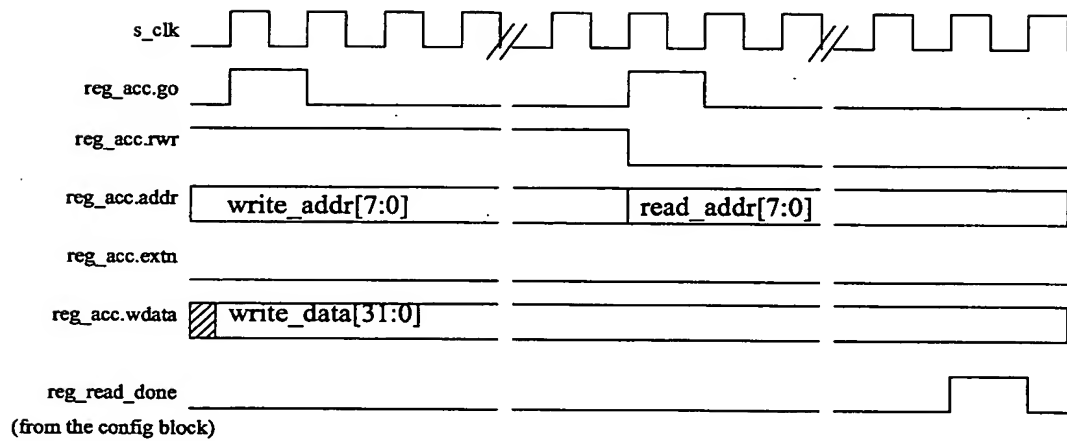


FIG. 181

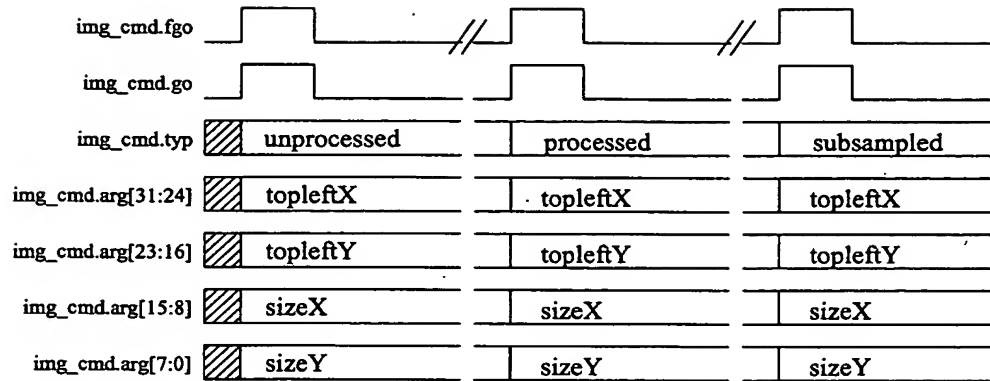


FIG. 182

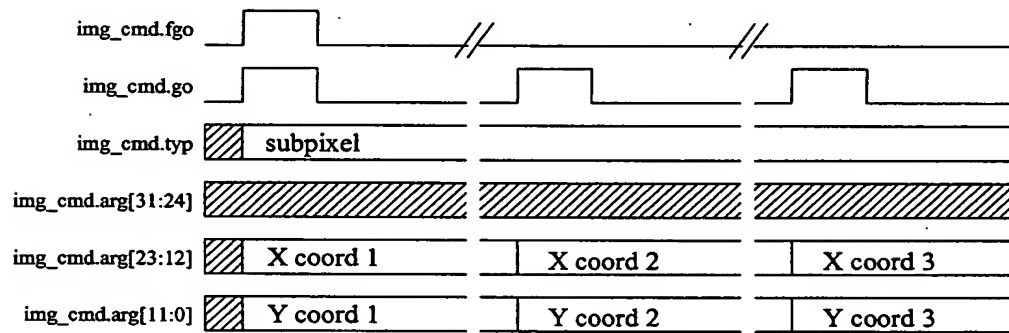


FIG. 183

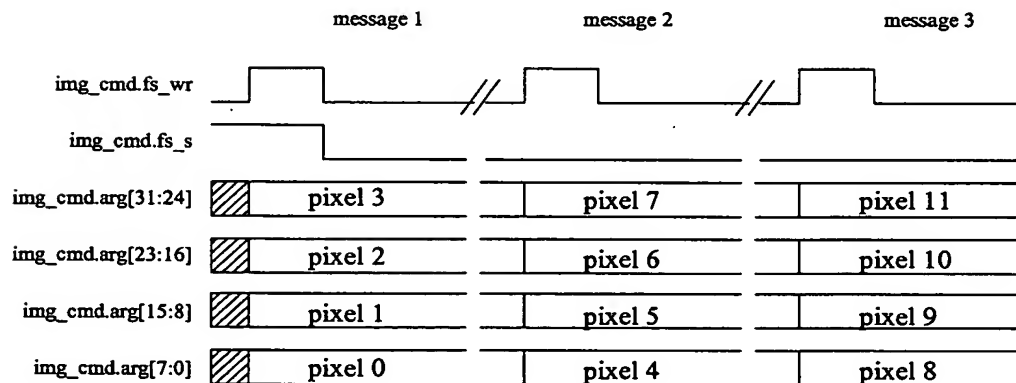


FIG. 184

132/157

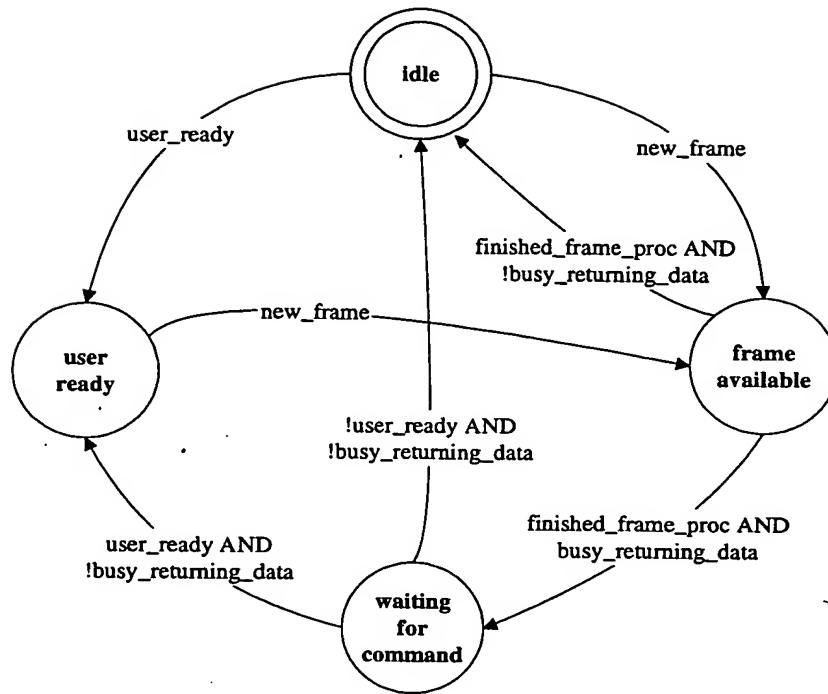


FIG. 185

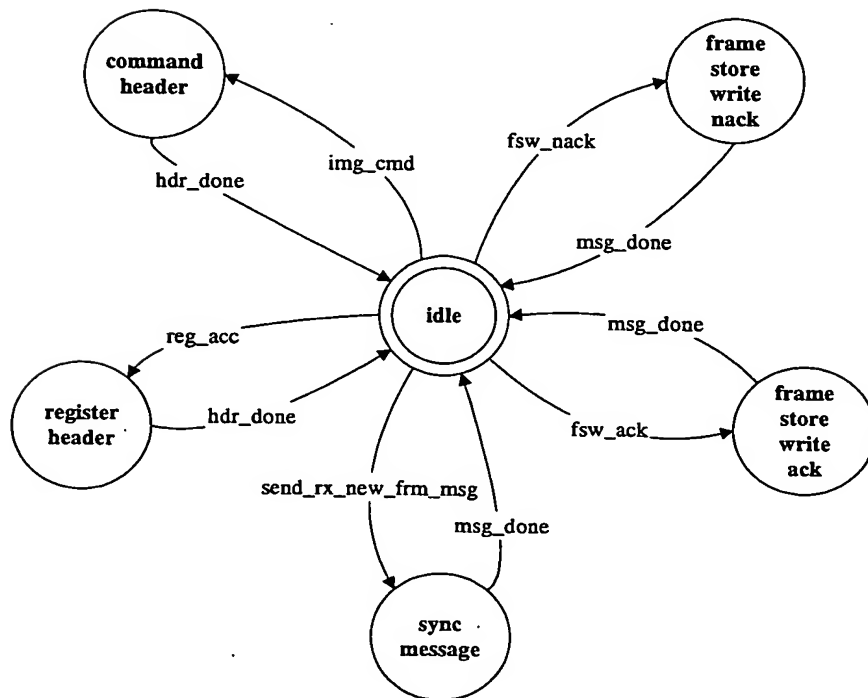


FIG. 186



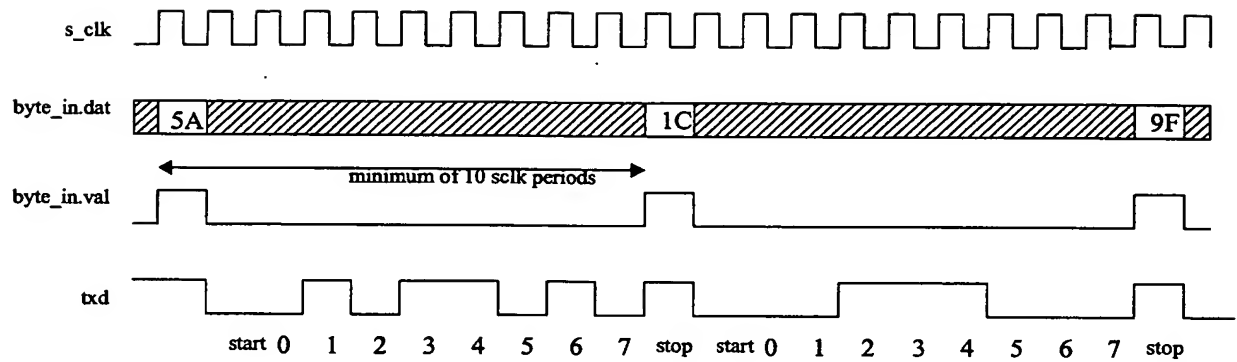


FIG. 187

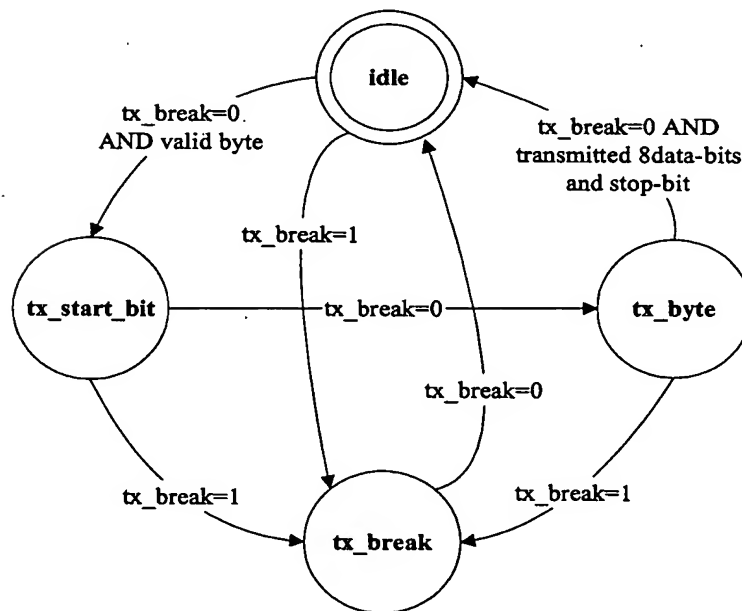
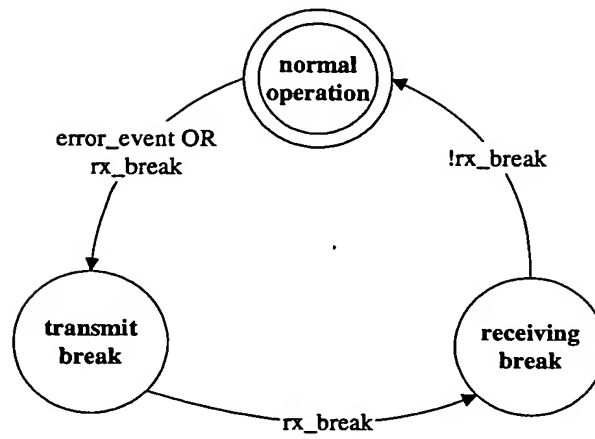


FIG. 188

*FIG. 189*

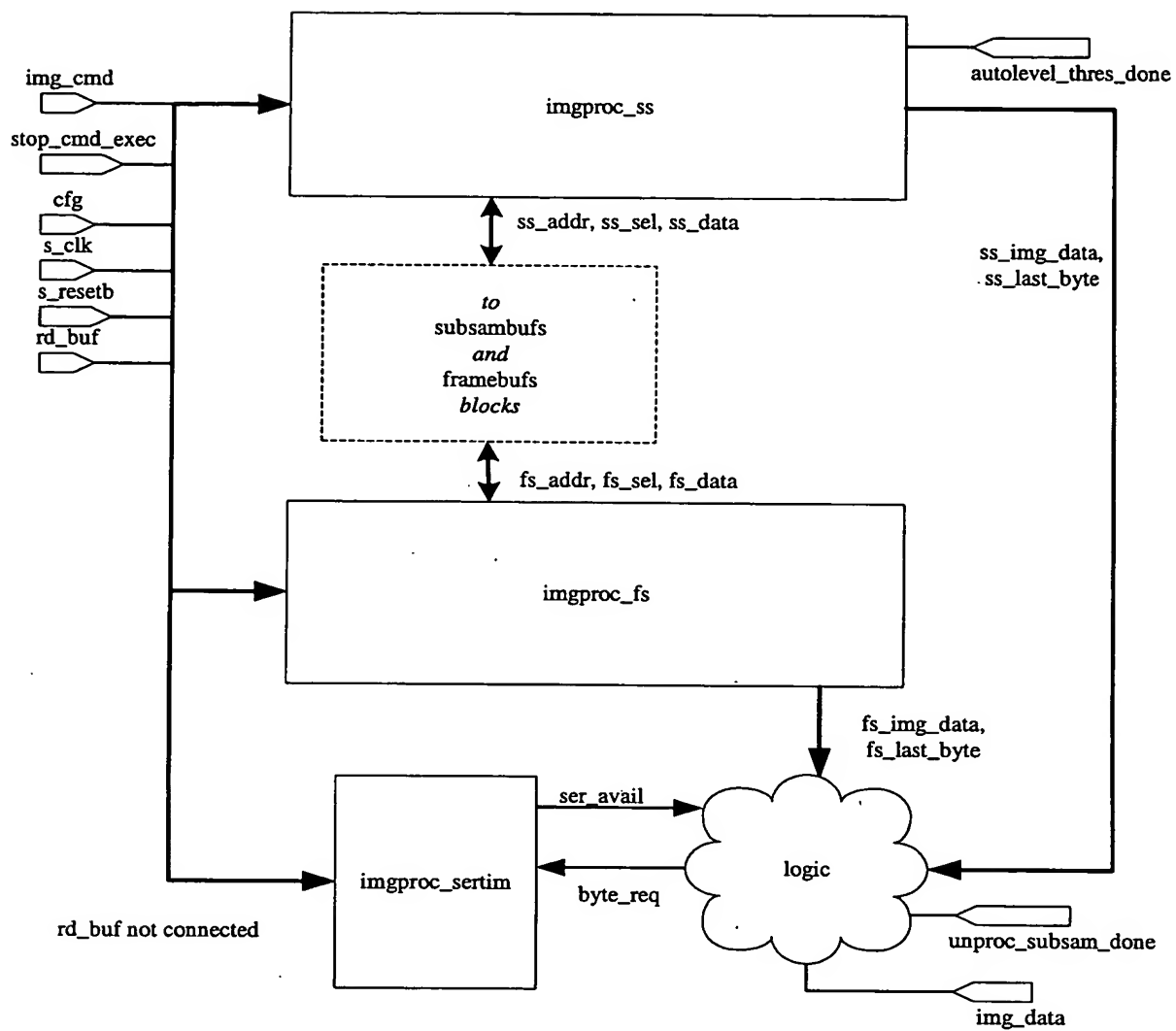


FIG. 190

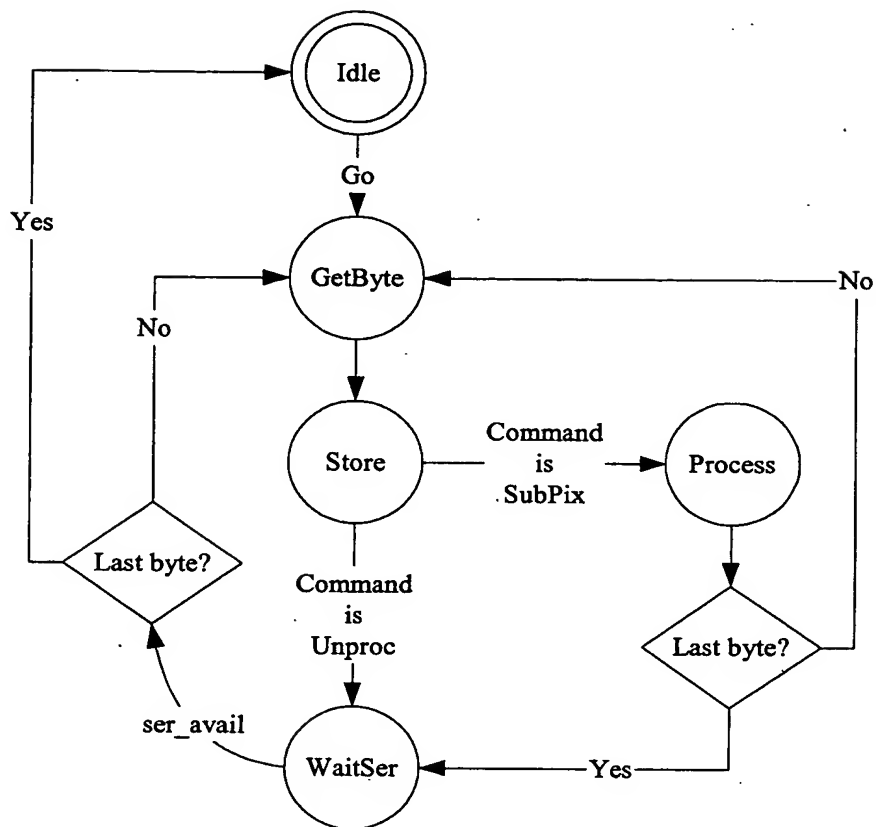


FIG. 191

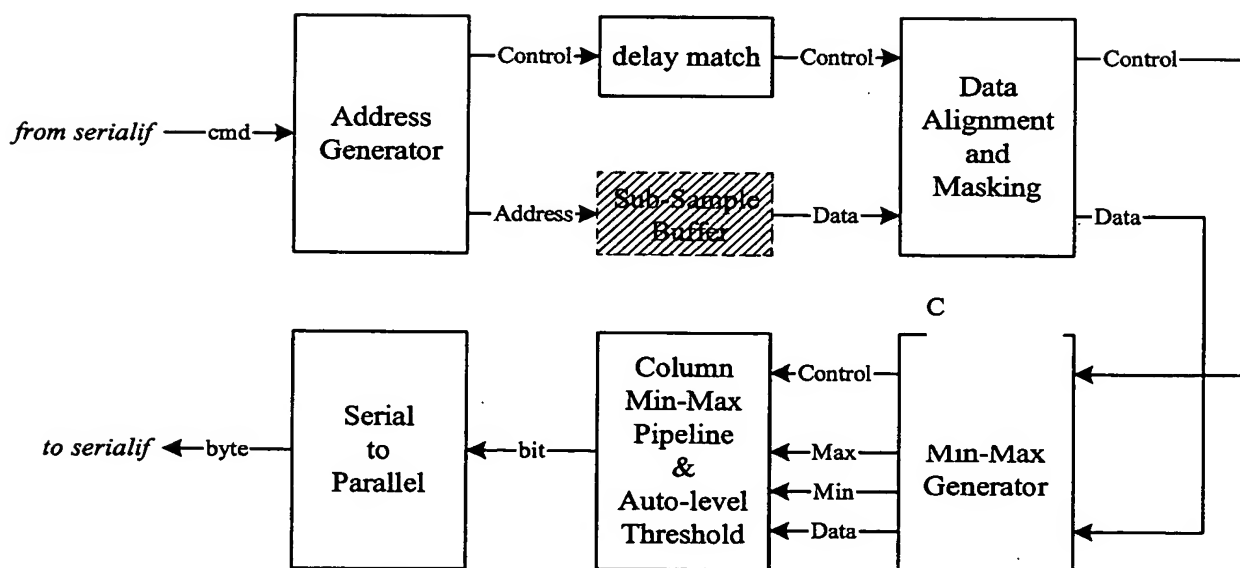


FIG. 192

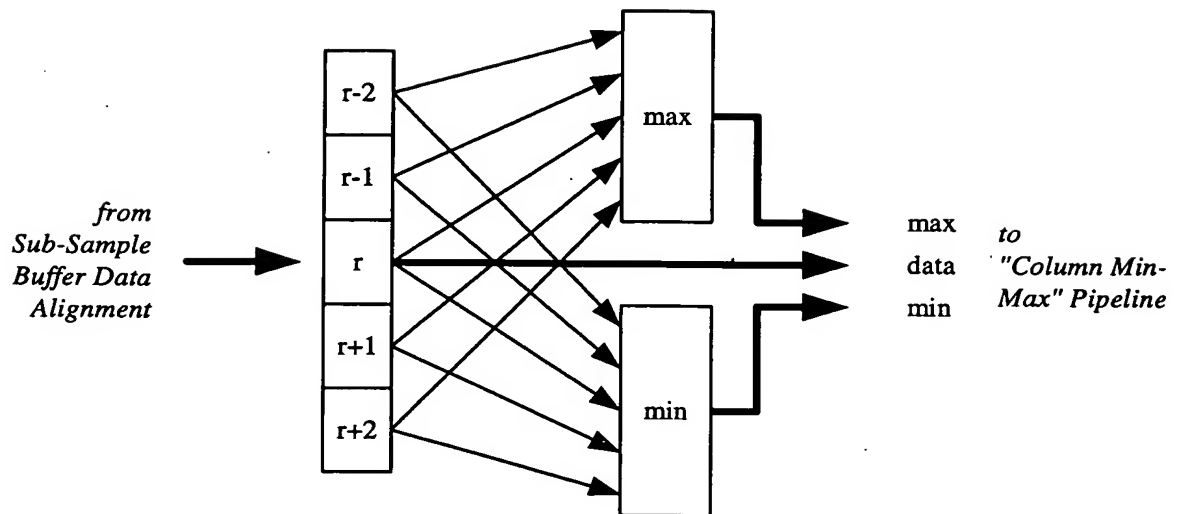


FIG. 193

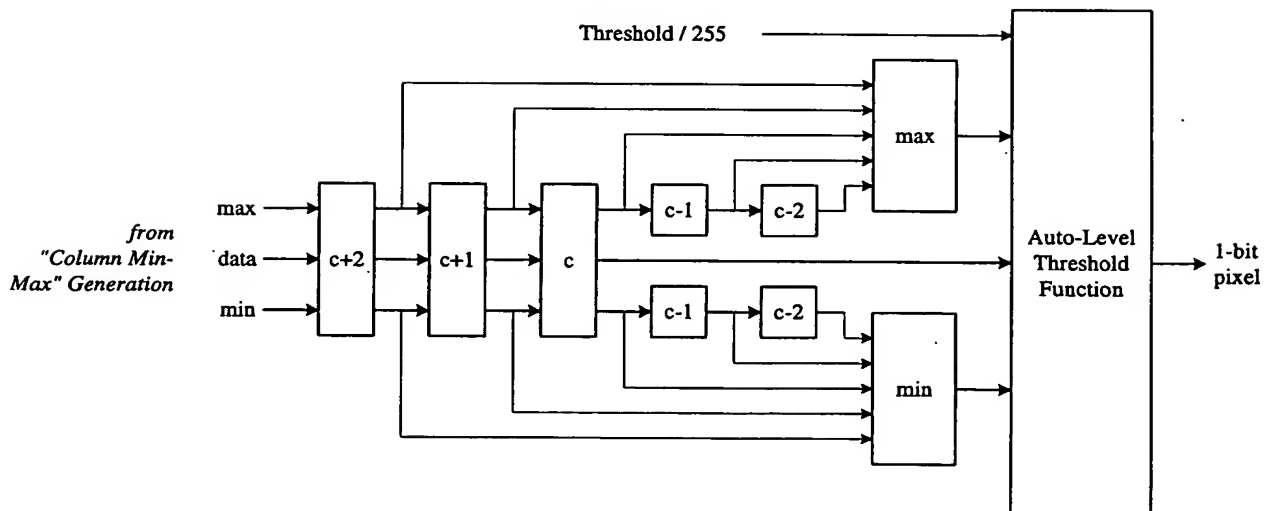


FIG. 194

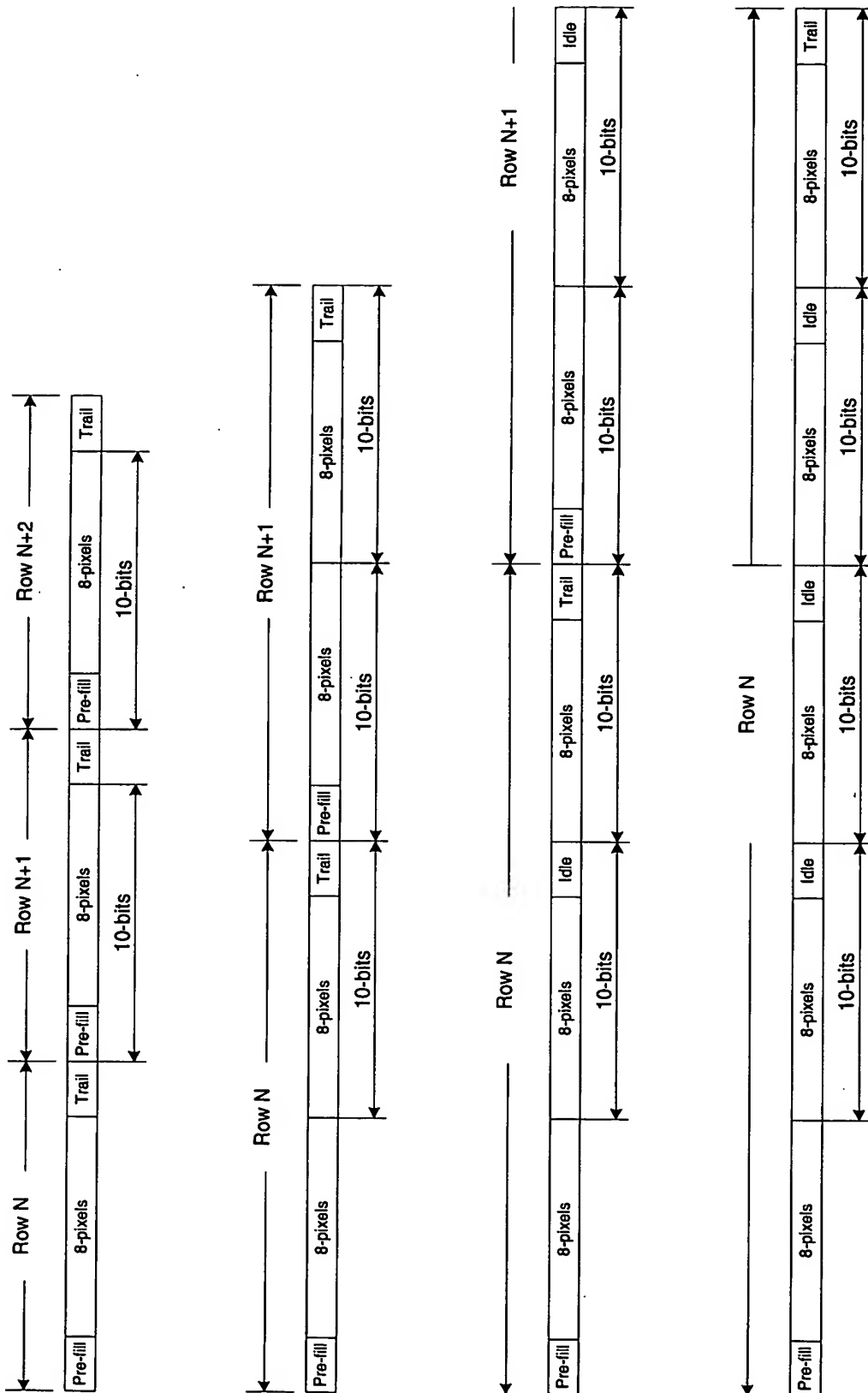
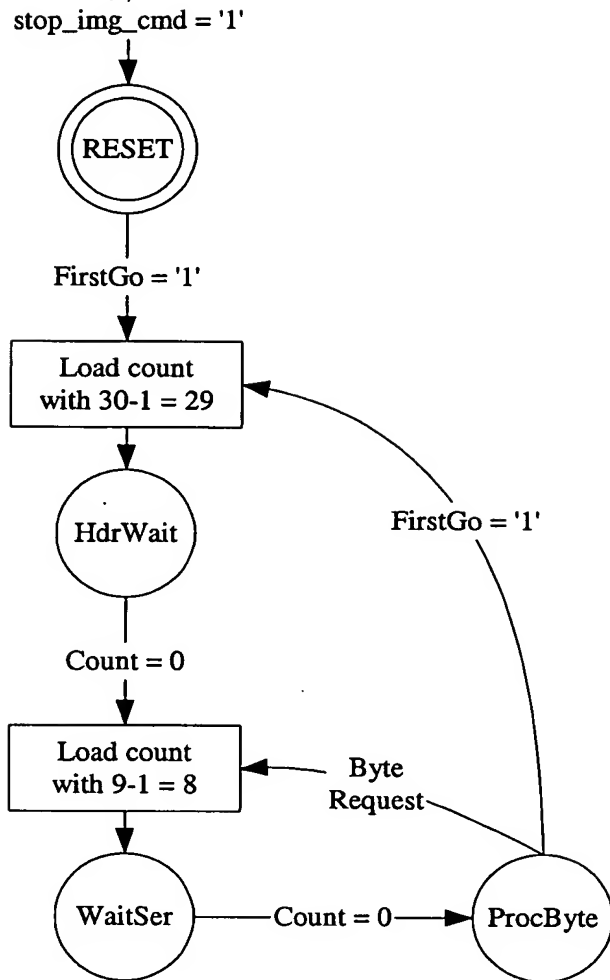


FIG. 195



## Notes:

## 1. FirstGo

This is the 'fgo' field of the image command from the serial\_if. This basically says - "Wait for the serial\_if to send out a command header before you start."

2. When `stop_img_cmd = '1'`, this acts as a global reset and overrides all other transitions.

3. The `ser_avail` output is '1' only during the ProcByte state. The ByteRequest may come immediately (in the same cycle), so this state may only last for one cycle.

4. The HdrWait state will last for 30 cycles. The WaitSer state will last for 9 cycles, and when added to the minimum one ProcByte state, we get the required 10 cycles for every byte.

FIG. 196

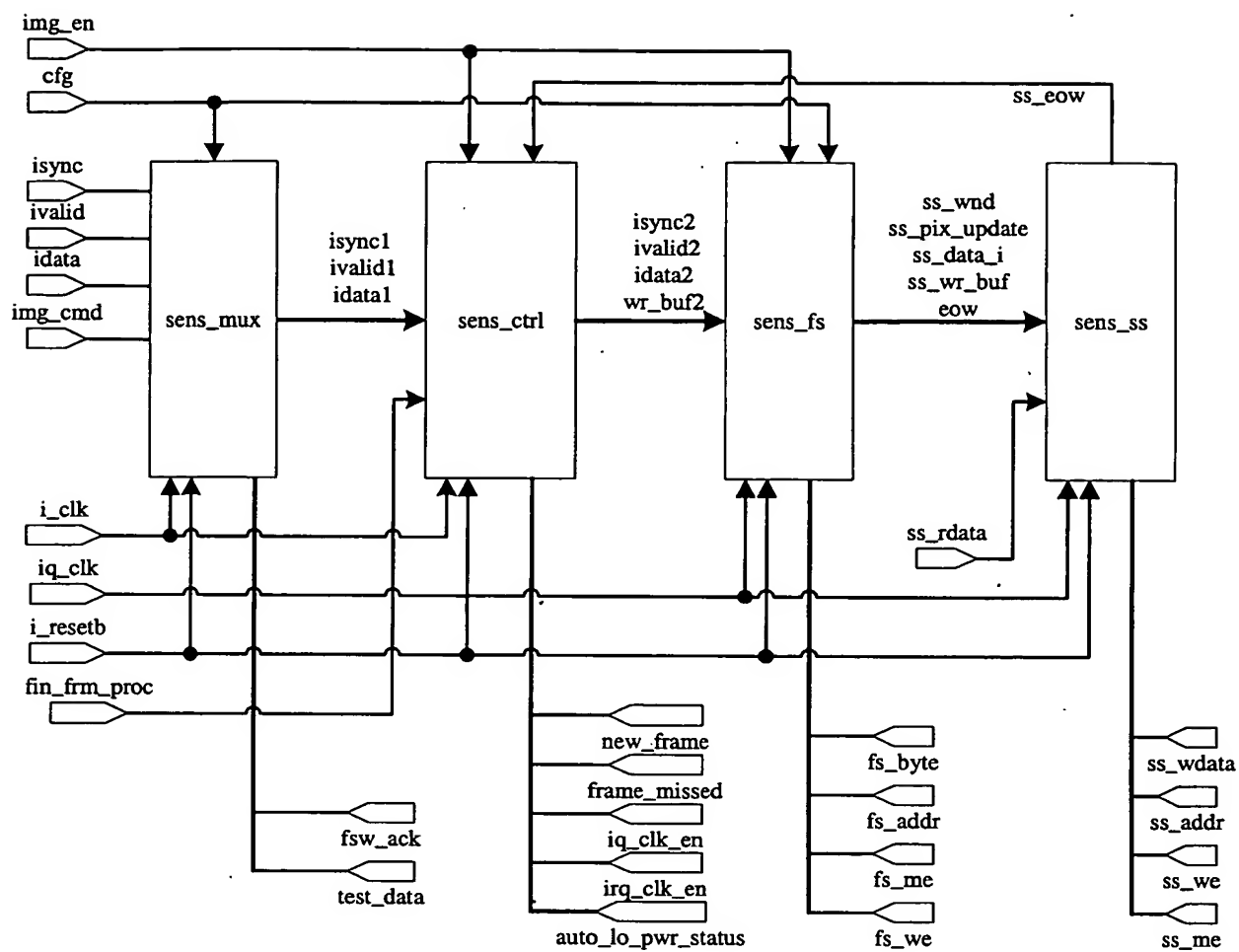
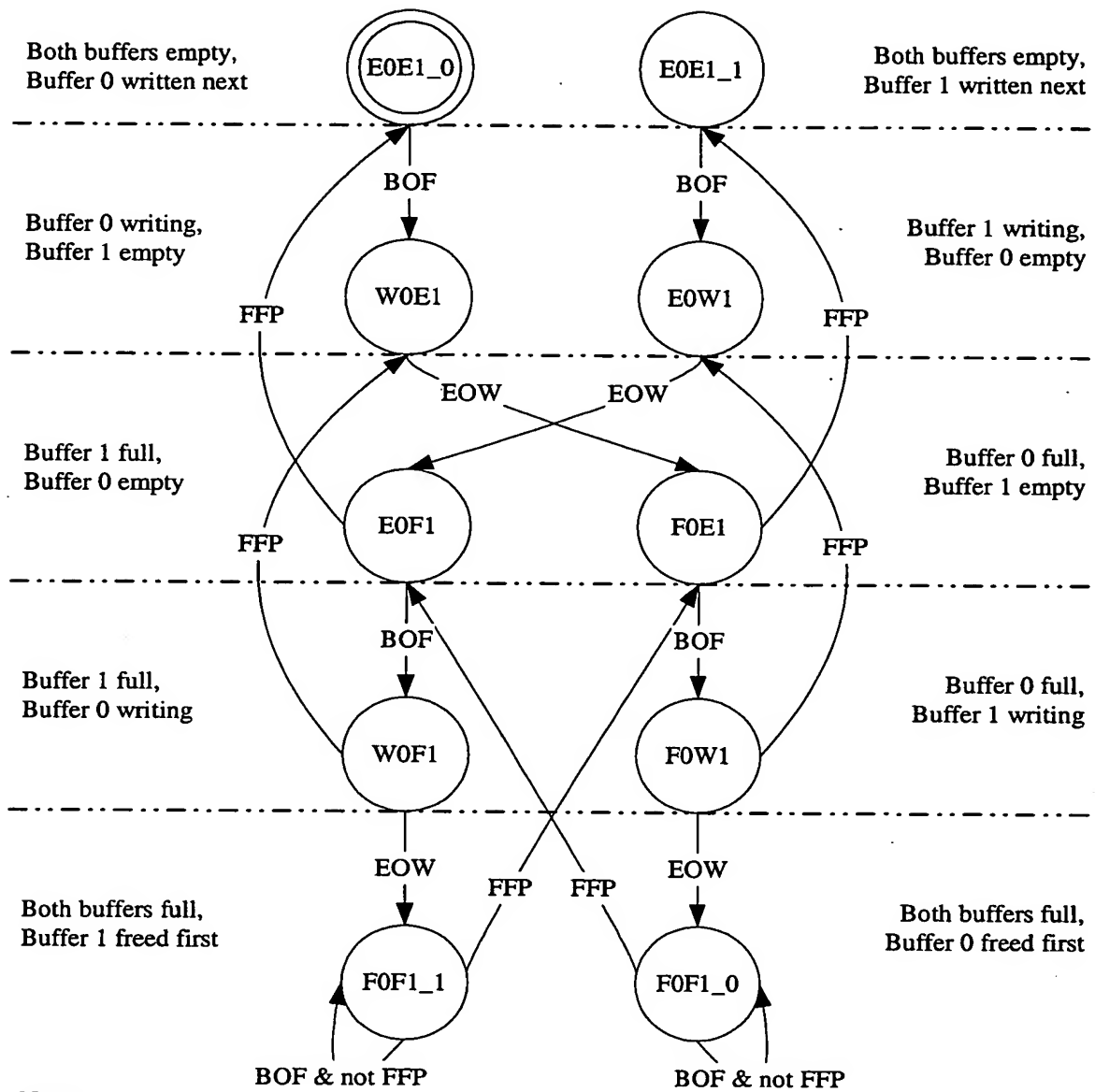


FIG. 197





## Notes:

1. BOF - Beginning Of Frame. isync1 pulse from sens\_mux
2. EOW - End Of Window from sens\_ss
3. FFP- Finished Frame Processing event from the serialif

4. State names are coded to show the state of both buffers:

E - empty  
F - full  
W - writing

When both are empty or full, the extra number on the end shows which will be written to next.

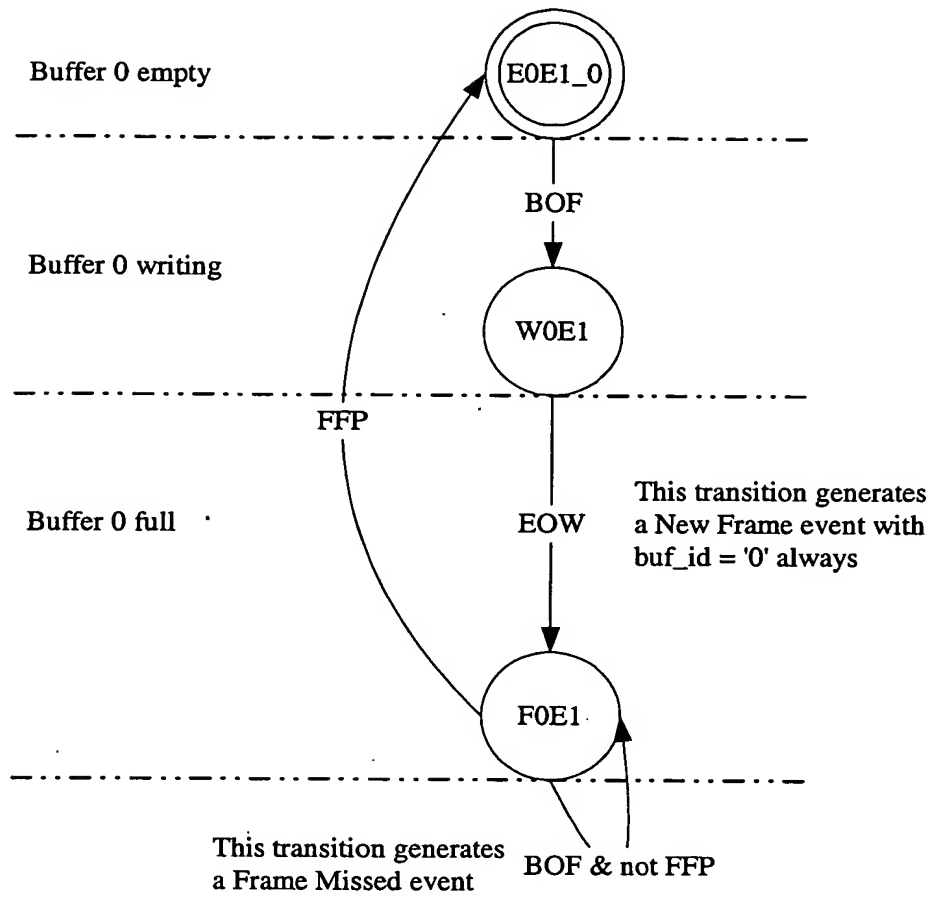
5. Transitions from one side to the other generate NewFrame events to the serialif. The NewFrame event includes the number of the buffer just filled.

6. Where the head of a FFP transition meets the tail of a BOF or EOW transition, the tail transition will be followed (if true) before entering the state.

For example:

From state E0F1, if FFP is true but BOF is false, E0E1 is the next state. If FFP is true and BOF is true, W0E1 is the destination.

FIG. 198

*FIG. 199*

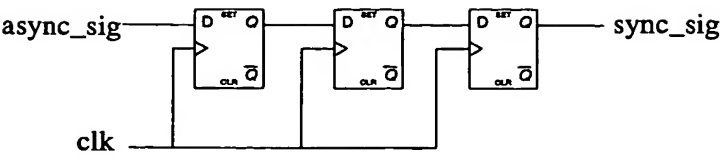


FIG. 200

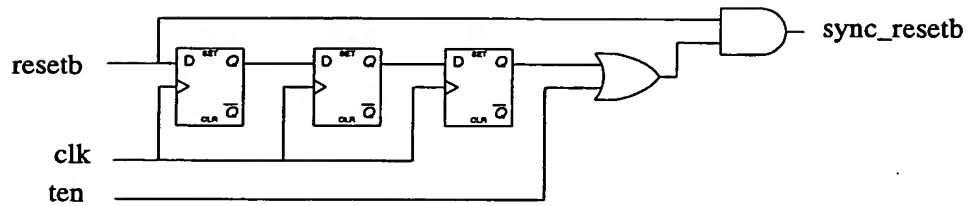


FIG. 201

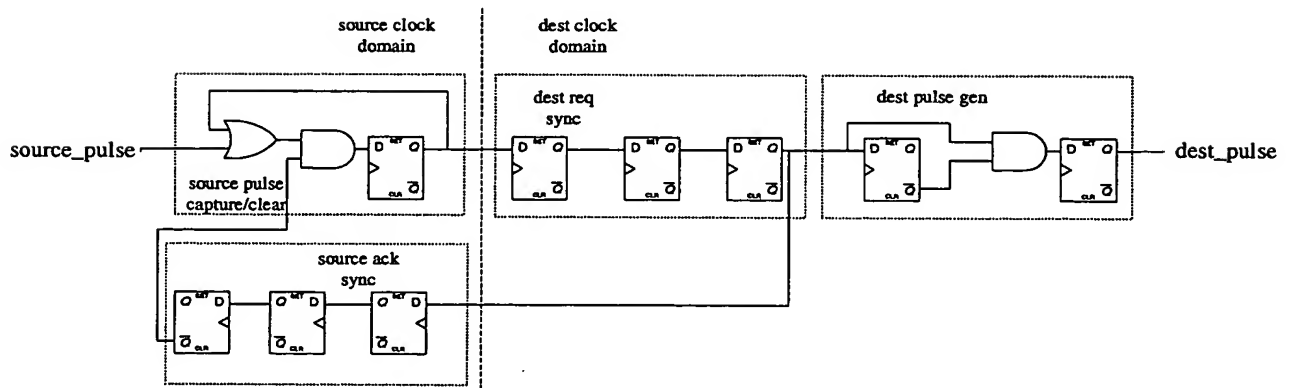


FIG. 202

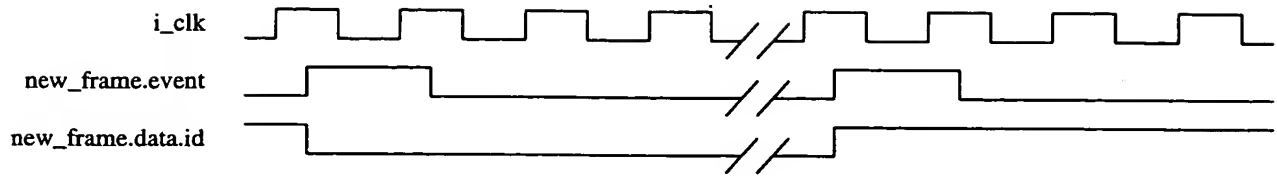


FIG. 203

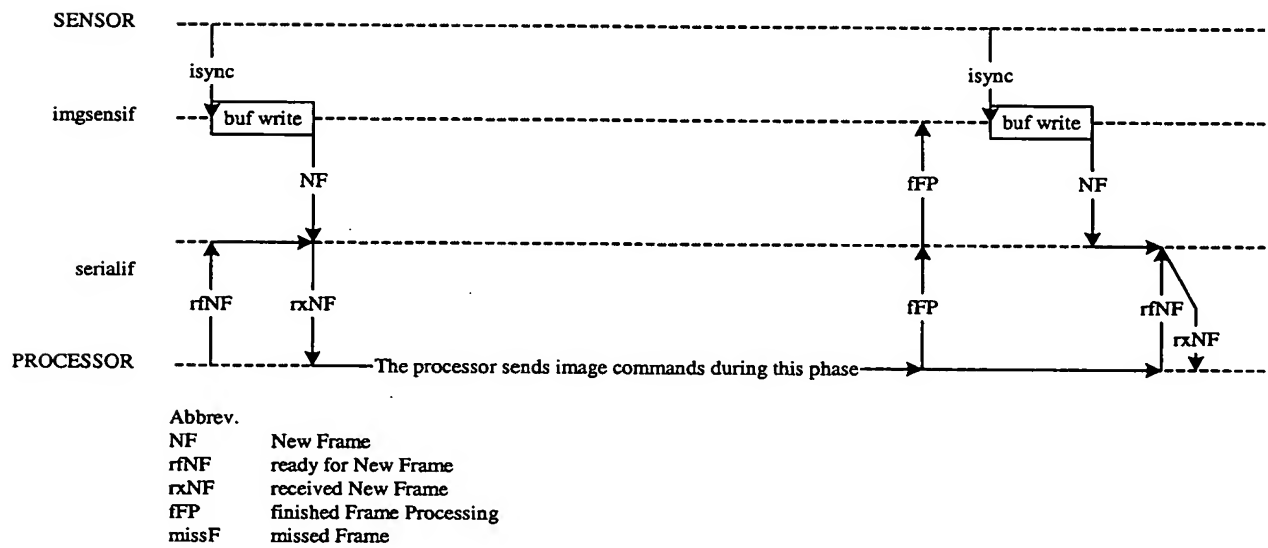


FIG. 204

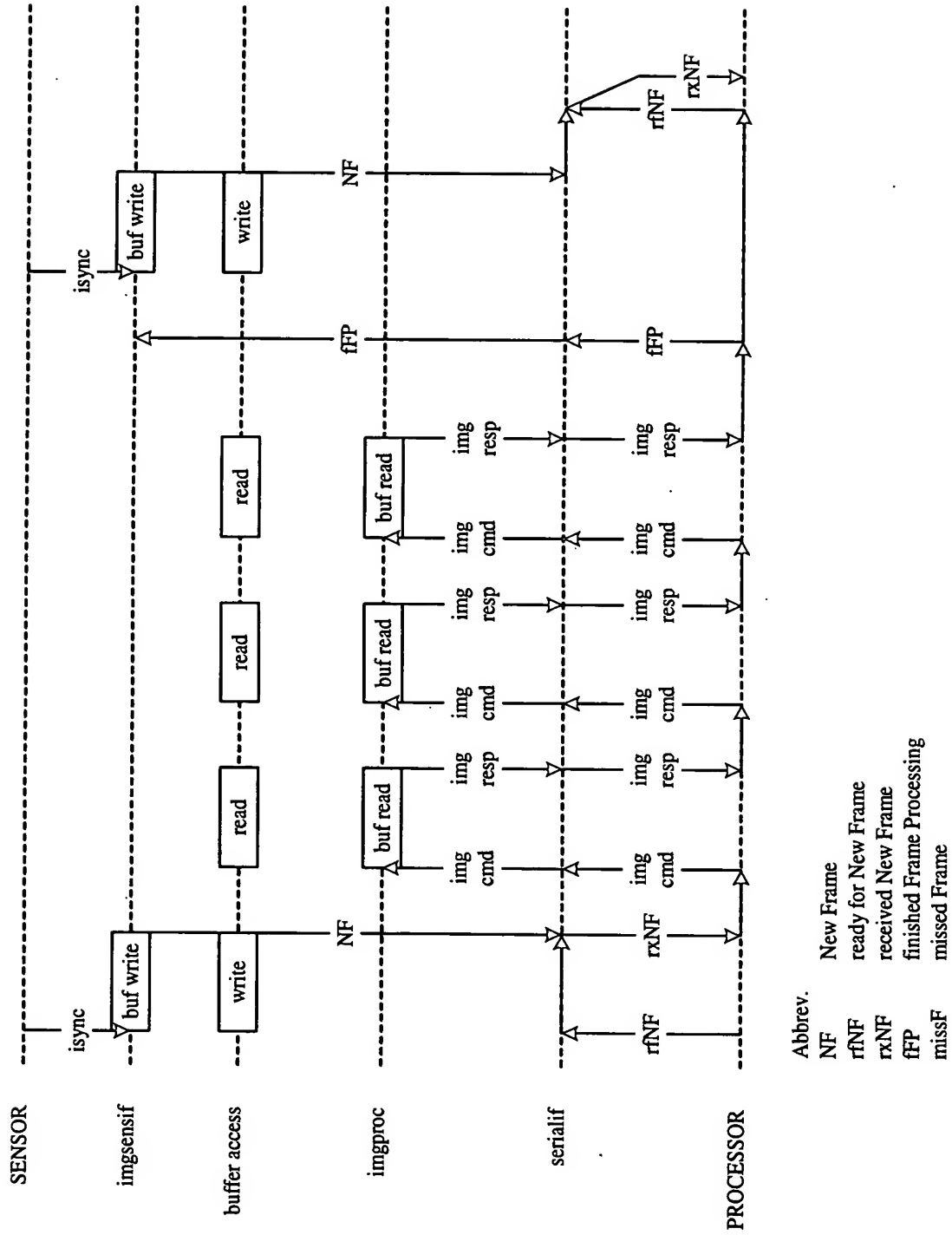


FIG. 205



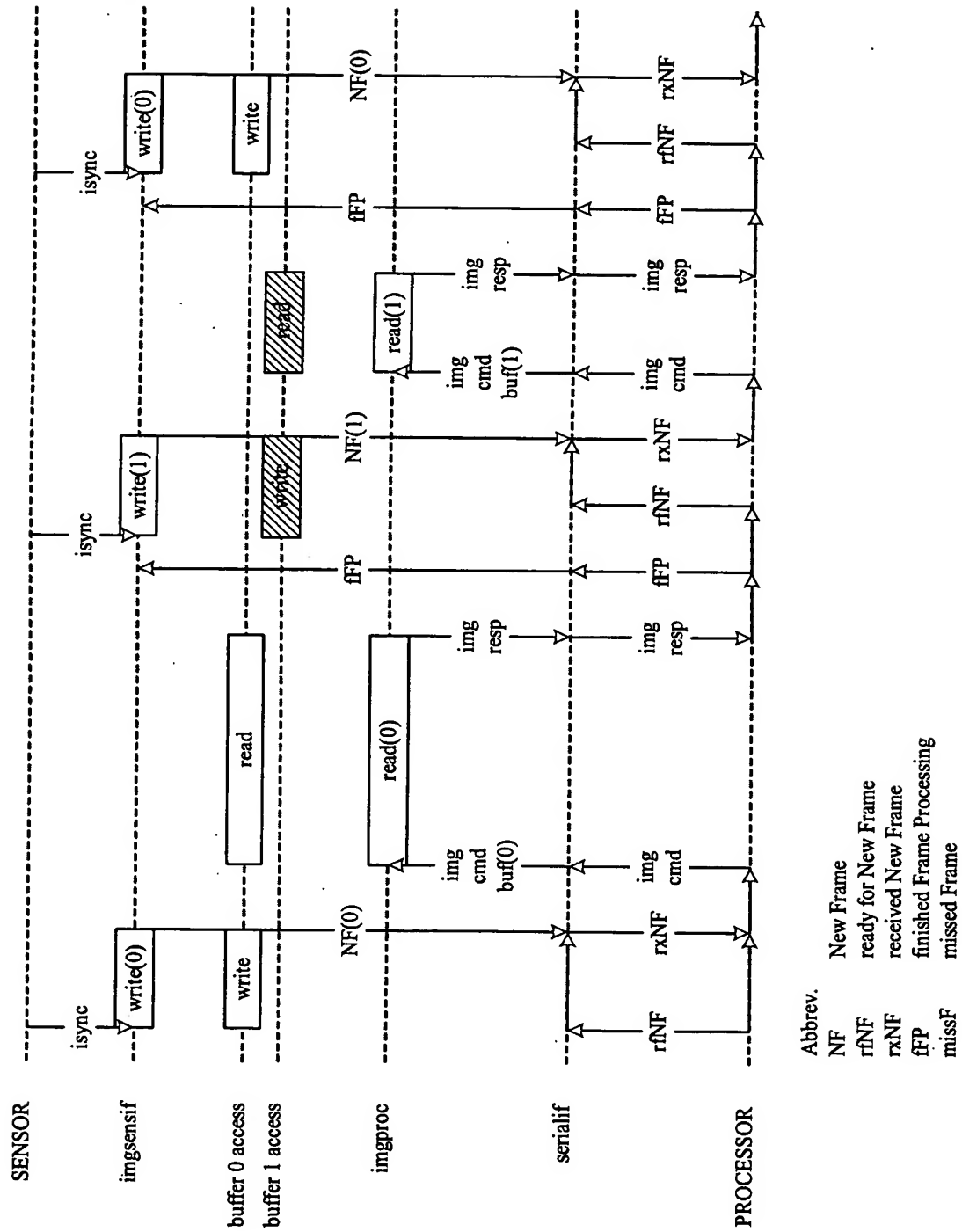


FIG. 207

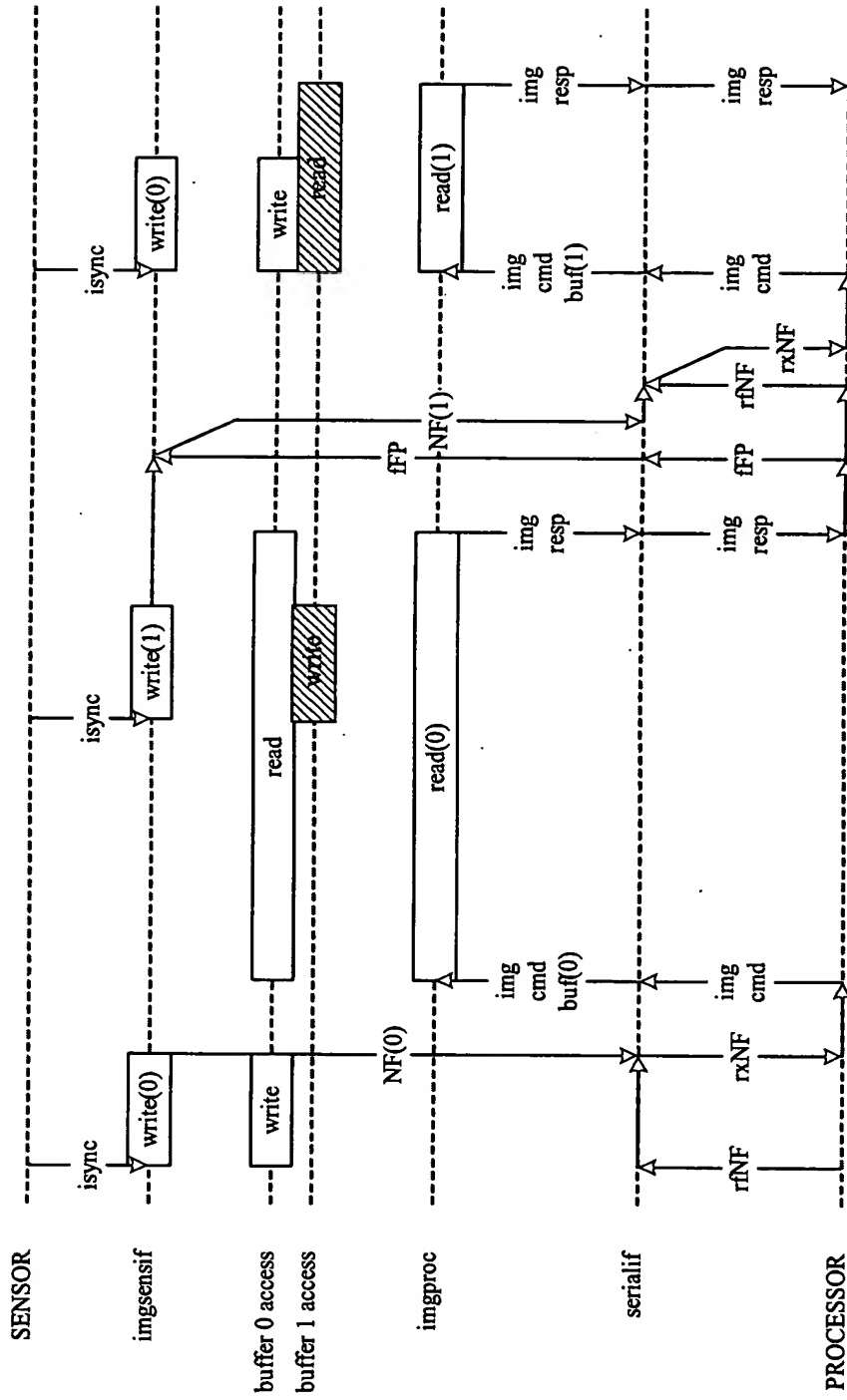


FIG. 208



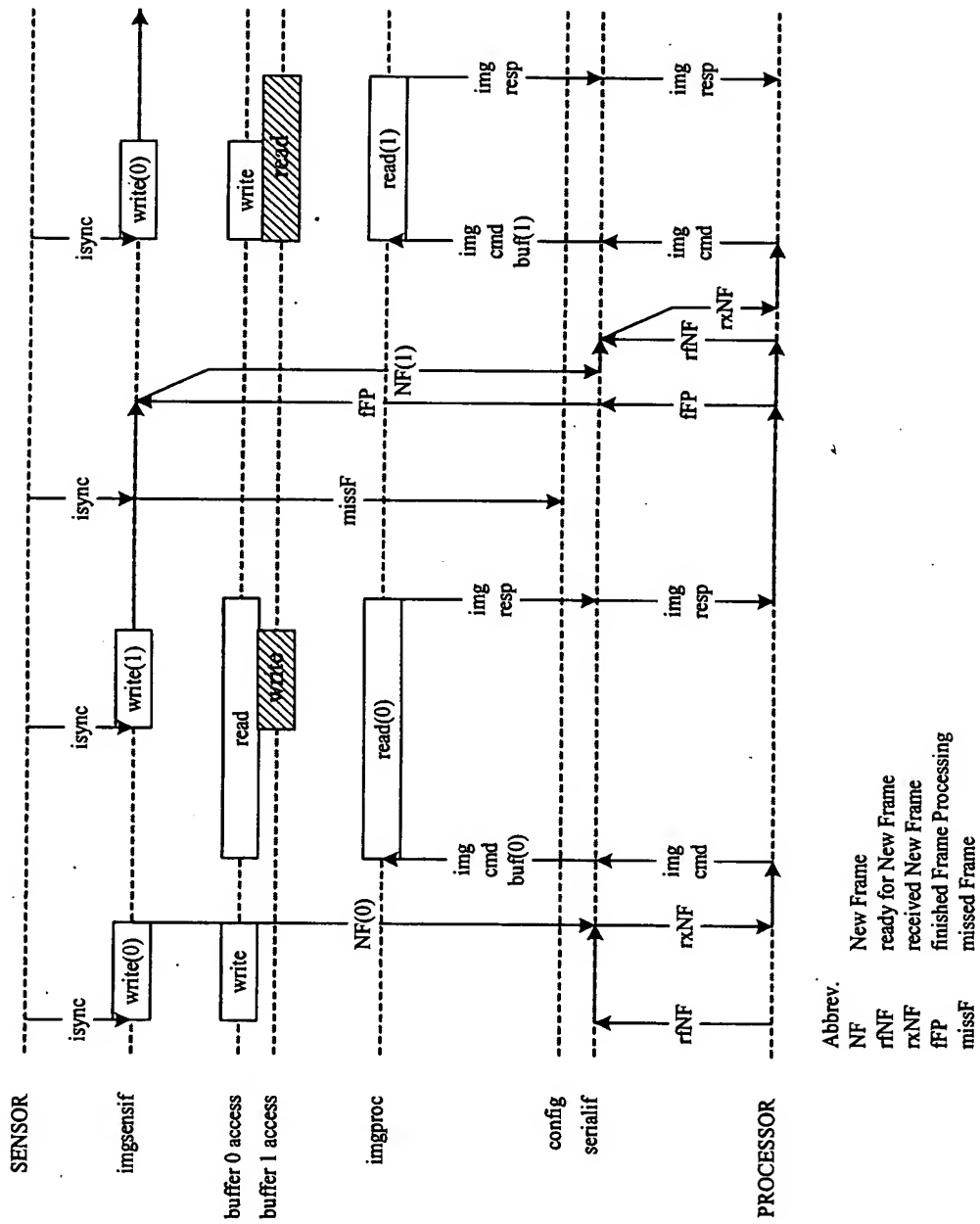


FIG. 209

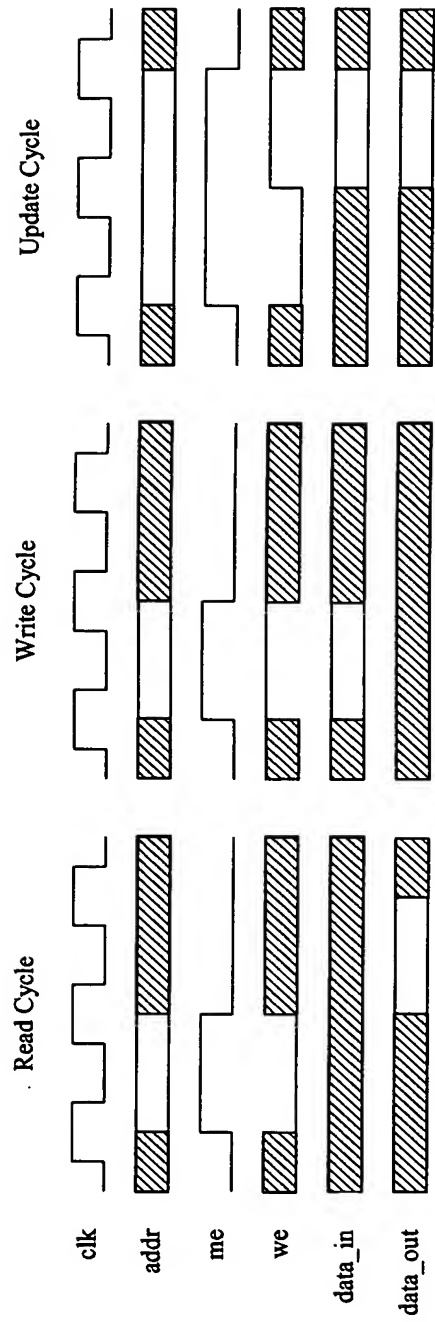


FIG. 210

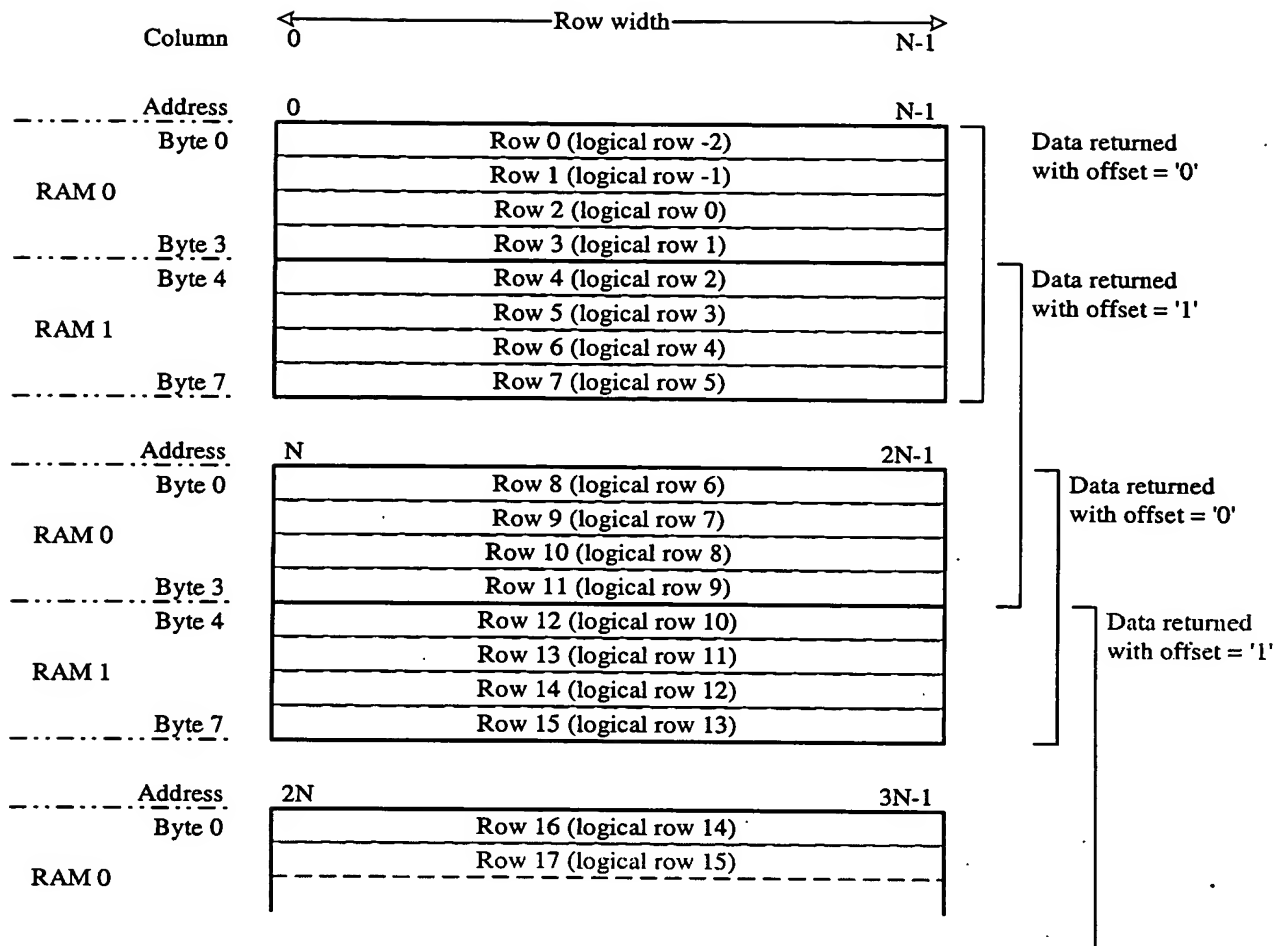


FIG. 211

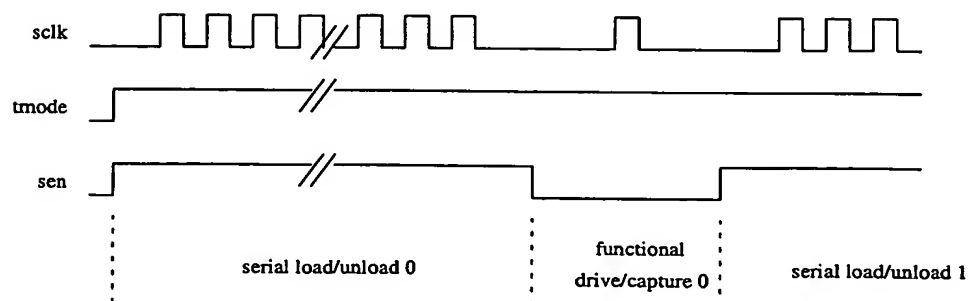


FIG. 212

152/157

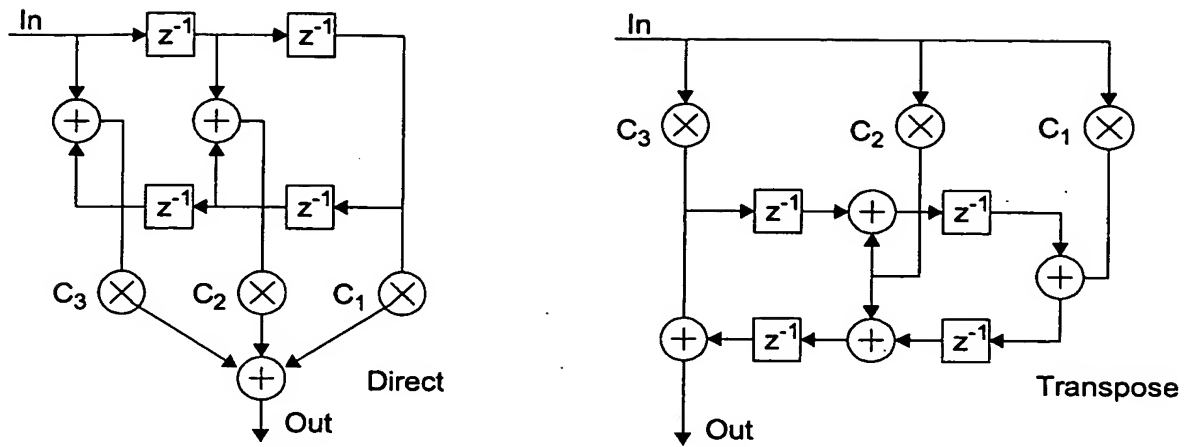


FIG. 213

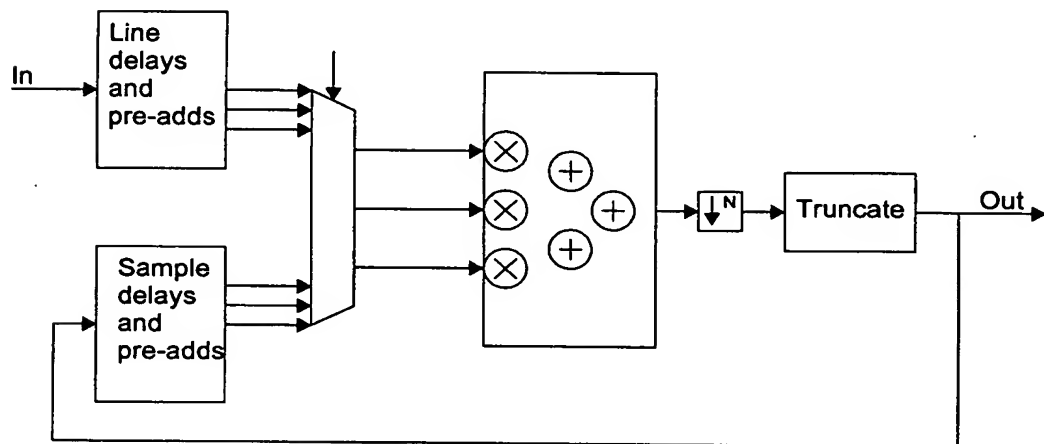


FIG. 214

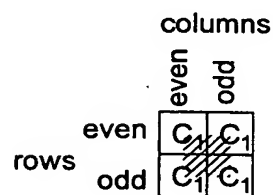


FIG. 215

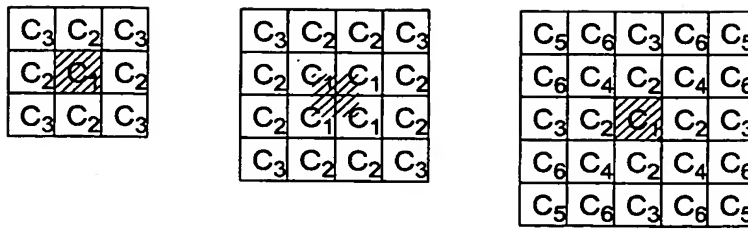


FIG. 216

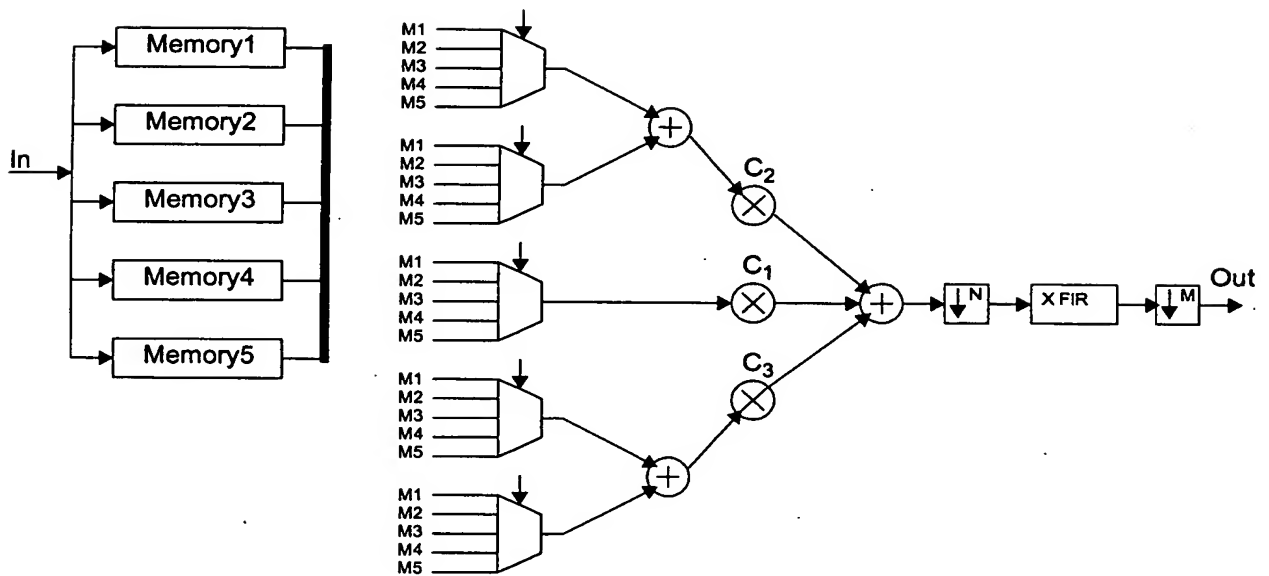


FIG. 217

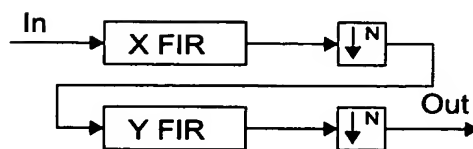


FIG. 218

154/157

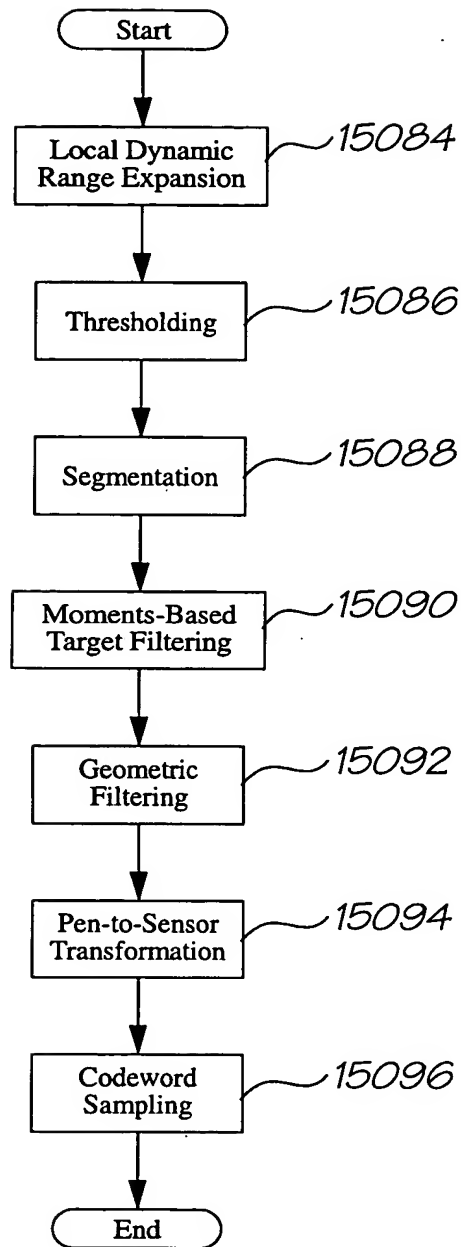


FIG. 219

155/157

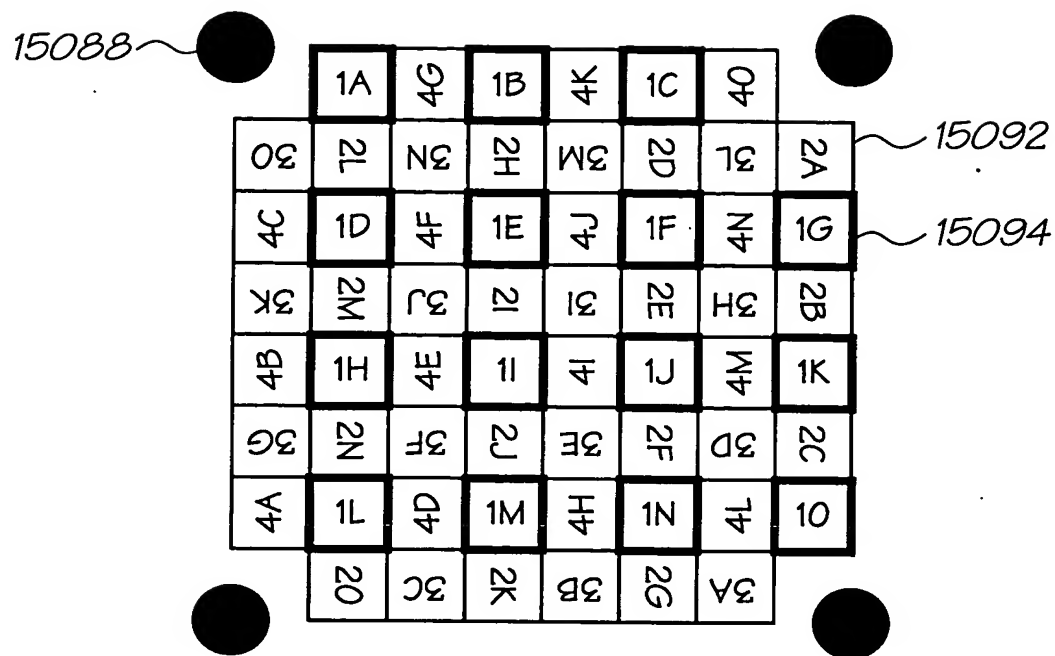


FIG. 220

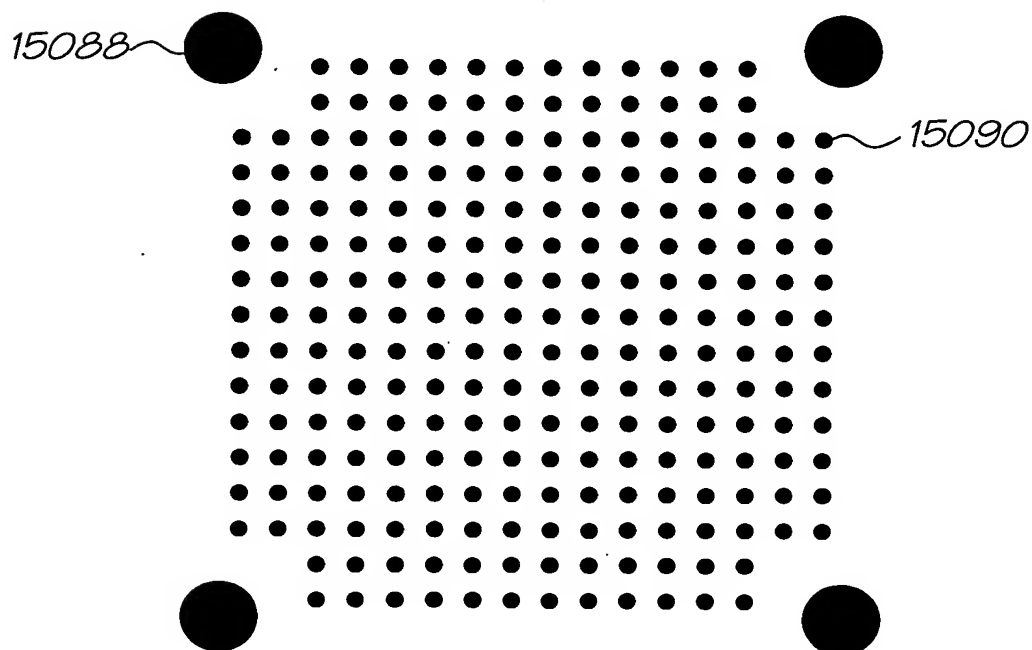


FIG. 221

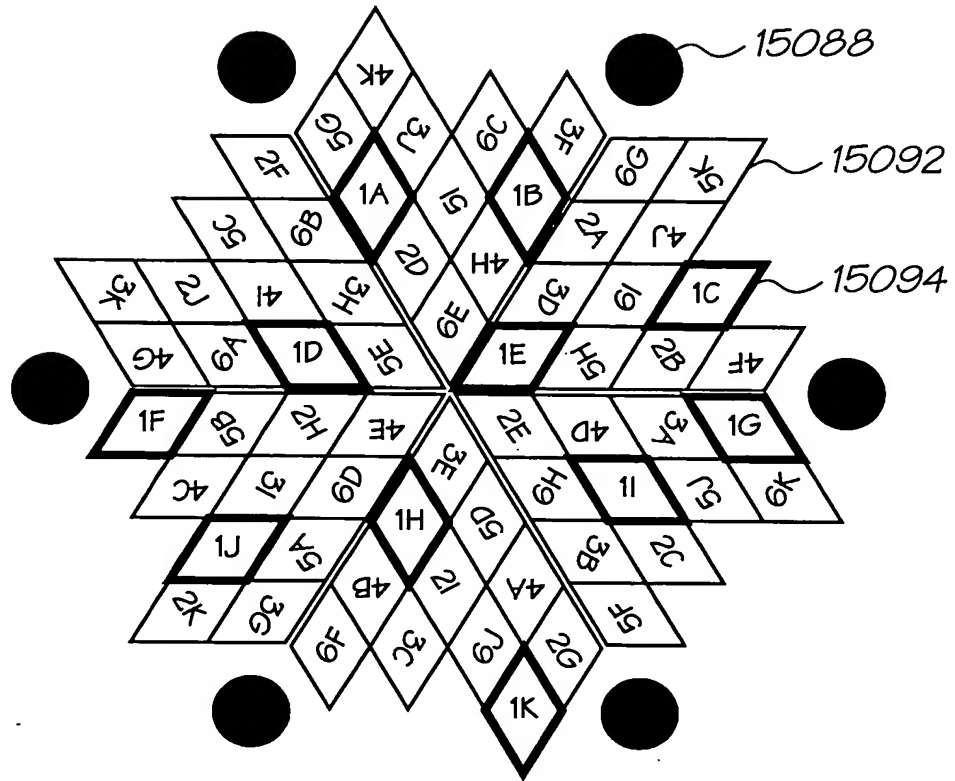


FIG. 222



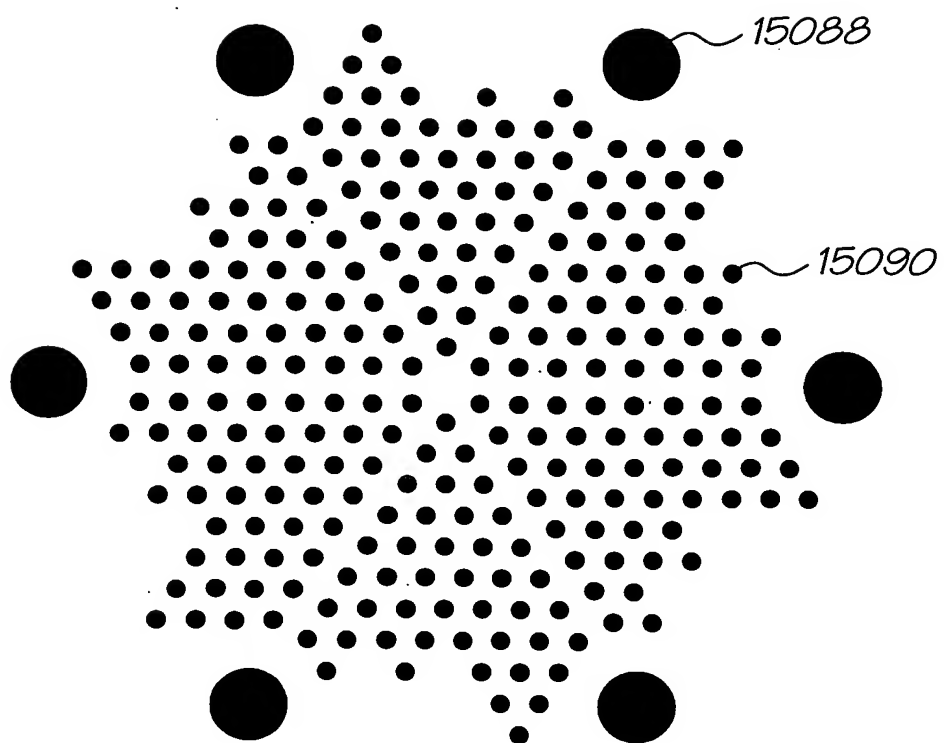


FIG. 223